

## **UNIVERSITÄT ROSTOCK**

FAKULTÄT FÜR INFORMATIK UND ELEKTROTECHNIK  
INSTITUT FÜR GERÄTESYSTEME UND SCHALTUNGSTECHNIK

## **DISSERTATION**

Selbstlernende akustische Mustererkennung zur Erfassung von  
Bauteilfehlern im Automobil

vorgelegt von: Stefan Twieg

geboren am: 31. Dezember 1981 in: Köthen (Anh.)

zum  
Erlangen des akademischen Grades

## **DOKTOR DER INGENIEURWISSENSCHAFTEN**

**- Dr.-Ing. -**

WISSENSCHAFTLICHE BETREUUNG:

Prof. Dr.-Ing. habil. Helmut Beikirch, Universität Rostock, IEF, IGS

Dr. rer. nat. Holger Opfer, Volkswagen Group Japan, VTT

URN: urn:nbn:de:gbv:28-diss2010-0122-3

1. Gutachter:

Prof. Dr.-Ing. habil. Helmut Beikirch  
Universität Rostock, Fakultät für Informatik und Elektrotechnik  
Institut für Gerätesysteme und Schaltungstechnik  
A.-Einstein-Str. 2, D-18051 Rostock

2. Gutachter:

Prof. Dr.-Ing. Matthias Sturm  
Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH)  
Fachbereich Elektrotechnik und Informationstechnik Institut für Nachrichtentechnik  
Postfach 30 00 66, 04251 Leipzig

3. Gutachter:

Prof. Dr.-Ing. Matthias Rötting  
Technische Universität Berlin, Fakultät für Psychologie und Arbeitswissenschaft  
Institut Mensch-Maschine-Systeme  
Franklinstr. 28-29, 10587 Berlin

Datum der Einreichung: 05.05.2010

Datum der Verteidigung: 20.09.2010

Veröffentlichungen über den Inhalt der Arbeit sind nur mit schriftlicher Genehmigung der Volkswagen AG zugelassen. Die Ergebnisse, Meinungen und Schlüsse dieser Dissertation sind nicht notwendigerweise die der Volkswagen AG

# Widmung

Diese Dissertation möchte ich meinem Vater Prof. Wolfgang Twieg<sup>†</sup> widmen,  
der mir diesen Weg ermöglicht hat.



## Kurzfassung

Die Akustik im Fahrzeuginnenraum steht in direktem Zusammenhang mit dem subjektiven Qualitätsempfinden. Hierzu bearbeitet die Konzernforschung der Volkswagen AG unter anderem Projekte zur Diagnose von Fahrzeuginnenraumgeräuschen. Das Gesamtgeräusch lässt sich als komplexe Summe diverser Schallereignisse betrachten. Durch ein geeignetes Mikrofon lässt sich das Innengeräusch erfassen, über eine Signalverarbeitung lässt sich das Signal einer Mustererkennung zuführen. Die Vision des Forschungsprojektes ist die Überwachung und selbstständige Analyse des Fahrzeugklangs im Innenraum auch während der Nutzungsphase durch den Kunden. Des Weiteren soll die Einhaltung der Vorgaben zum Erreichen eines qualitativ hochwertigen Gesamtfahrzeugklangs im Produktionsprozess überprüft und objektiv beurteilt werden. Abweichungen im Produktionsprozess und etwaige Kundenbeanstandungen, bei denen ein hervortretendes Schallereignis auf ein verschlissenes oder defektes Bauteil hinweist, können derzeit nicht durch automatisierte Verfahren analysiert und beurteilt werden.

Neben der Durchführung von Messungen an Versuchsfahrzeugen und dem Datenbankaufbau sind parallel Berechnungsmodelle für den Mustererkennungsprozess entwickelt und umgesetzt worden. Grundlage für diese Algorithmenentwicklung bildet die automatisierte Berechnung zustandsbeschreibender Kenngrößen aus den Zeitsignalen. Eine herausfordernde Aufgabe ist die Verknüpfung objektiver Messgrößen mit dem subjektiven Geräuschempfinden, welches über psychoakustische Parameter abgebildet wird.

Die Schwerpunkte dieser Promotion, die in der Volkswagen Konzernforschung in Zusammenarbeit mit dem Institut für Gerätesysteme und Schaltungstechnik der Universität Rostock angefertigt wurde, liegen in der Konzept- und Algorithmenerstellung zur automatisierten Analyse des Fahrzeuginnenraumgeräusches mit Anwendung in der Qualitätssicherung, der Produktion und den Werkstätten (siehe Patentschrift [Opf09]). Die Aufnahme akustischer Kenngrößen in alltäglicher Umgebung, die Datenbankablage und die Kenngrößenberechnung stellt eine Grundlage für den Algorithmenentwurf dar. Über neuronale Algorithmen (selbstorganisierende Karten) ist das Ziel erreicht worden, zeitabhängige akustische Größen anhand berechneter Kenngrößen zu interpretieren, wodurch eine Unterscheidung von Fehlerzuständen ermöglicht wird. Für die Umsetzung der Erkenntnisse sind zum einen ein Werkstatt-Diagnosetester zur akustischen Analyse am Kundenfahrzeug und zum anderen das *mobile akustische Diagnose-System* (mADS) für den Einsatz in Produktion und Qualitätssicherung entwickelt worden.



## Summary

Interior vehicle acoustic is in close connection with our quality impression. The Volkswagen Group Research is working on a project called Acoustic Diagnostic System.

The interior vehicle noise is recorded by microphones. This recorded noise can be considered as a sum of various sound emission sources. Thus, the noise can be processed in the next step by signal processing and pattern recognition to identify the sound source.

The major subject of this research project is the monitoring and analysis of vehicle sound during the vehicle life cycle.

Another point is the possibility to verify that the high sound quality defined in the development process is met at the end of the production line by means of objective methods. Currently, an automated objective method of interior vehicle noise judgment for deviance in production process and customer service requests, where e. g. rattling or squeaking points to a damaged or worn-out part, is not available.

Next to the measurement of research vehicles and database development the concept and simulation of different pattern recognition methods is done. Algorithms are developed for automated calculation of condition specific parameters (physical and psychoacoustical) for recorded microphone time signals. Especially the analysis of the connection between these parameters and the subjectiv human sense of hearing is another interesting topic.

The main focus of this doctoral thesis at the Volkswagen Group Research and the Institute of Electronic Appliances and Circuits of University Rostock is on the development of concepts and algorithms for automatic analysis of interior vehicle sound with applications in quality control, production and dealer workshops. The main target, to calculate and analyse the parameters, whether a normal interior sound or a noticeable noise occurs, is reached by neural network algorithms. Self-Organizing Maps are used for the interpretation of acoustic patterns of two research vehicles. Furthermore the knowledge of this doctoral thesis lead to the development of two measurement systems. The first one is an enhanced diagnostic device which can be used for acoustic analysis of customer vehicles and the second one is the *mobile acoustic diagnostic system* (mADS) for research, quality control and end of production line.





# Inhaltsverzeichnis

<b>Symbolverzeichnis</b>	<b>VII</b>
<b>Tabellenverzeichnis</b>	<b>IX</b>
<b>Abbildungsverzeichnis</b>	<b>XI</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Stand der Technik</b>	<b>3</b>
2.1 Akustische Kamera, Sprechererkennung & Spracherkennung, Sprachverbesserung	3
2.2 Product Life-cycle Management (PLM)	4
2.3 Akustische Analysen zur Qualitätssicherung	5
2.4 Prüfsystem-Merkmale im Überblick	10
2.5 Interpretation hochdimensionaler Daten	11
2.6 Ausrichtung der Arbeit	13
<b>3 Aufgabenstellung und Grundlagen der akustischen Mustererkennung</b>	<b>15</b>
3.1 Konzepterstellung	15
3.2 Mustererkennungsmodell	18
3.3 Überblick zum Aufbau der Arbeit	19
<b>4 Merkmalreduktion und Klassifikation</b>	<b>23</b>
4.1 Physikalische Kenngrößen als Merkmale	23
4.2 Psychoakustische Kenngrößen als Merkmale	25
4.3 Merkmalselektion: Erzeugung akustischer Fingerabdruck	27
4.3.1 PCA-Hauptkomponentenanalyse	30
4.3.2 Fisher Diskriminante	30
4.3.3 F-Ratio	31
4.3.4 Korrelationskoeffizient	31
4.3.5 Relief-Algorithmus für Mehrklassenanwendung	32
4.3.6 Informationsgewinn und Transinformation	32
4.3.6.1 Kullback-Leibler-Divergenz, Informationsgewinn	33
4.3.6.2 Transinformation, gegenseitige Information	34
4.3.6.3 Symmetric-Uncertainty/ Symmetrische Unbestimmtheit und Redundanz	35
4.3.7 Fast Correlation Based Filter - FCBF	35
4.4 Mustererkennung zur Fehlergeräuscherkennung	37

4.4.1	Unterteilung der Datenmenge in Test-/ Training-/ Validierungsmenge	38
4.4.2	Distanzmaße zur statistischen Mustererkennung	40
4.4.3	Klassifikatoren auf neuronaler Basis	42
4.4.3.1	Neuronenmodelle und Netztopologien	43
4.4.3.2	Wichtige Lernalgorithmen für Feedforward-Netze	44
4.4.4	Self-Organizing-Maps	45
4.4.5	Transparente recheneffiziente Klassifikationsalgorithmen	54
<b>5</b>	<b>Berechnung des akustischen Fingerabdrucks</b>	<b>59</b>
5.1	Beschreibung der Datensätze	59
5.2	Normierung der Merkmale	63
5.3	Anwendung der Selektionsverfahren	64
5.3.1	Bewertung der Selektionsverfahren	64
5.3.2	Fisher-Diskriminante	67
5.3.3	Fisher-Diskriminante und Korrelationskoeffizient	71
5.3.4	Fisher-Diskriminante und Relief	75
5.3.5	Fisher-Diskriminante und Transinformation	76
5.3.6	Fisher-Diskriminante und Informationsgewinn	78
5.3.7	Fisher-Diskriminante und FCBF	79
5.3.8	Fazit	81
5.4	Fahrzeugspezifische akustische Fingerabdrücke im Vergleich	82
<b>6</b>	<b>Agent zur Fehlerindikation</b>	<b>87</b>
6.1	Korrelationsmethode zur Trennung akustischer Zustände	87
6.1.1	Einführung	87
6.1.2	Modellentwurf	88
6.1.2.1	Vorverarbeitung und Datenbank	89
6.1.2.2	Korrelationskennlinien	89
6.1.2.3	Mathematische Grundlagen	90
6.1.3	Algorithmus	91
6.1.4	Test des Verfahrens und Auswertung	92
6.1.5	Auswertung der Korrelationsmethode	94
6.2	Überwachte akustische Diagnose mit Self-Organizing-Maps	95
6.2.1	Theorie und Hintergrund	95
6.2.2	SOM zur akustischen Zeitsignalbewertung und Fehlerdiagnose	96
6.2.3	Test und Funktionsnachweis	102
6.2.4	Modellparameter und Optimierungen	106
6.2.5	Erkenntnisse	116
6.3	Analyse der Hörbarkeit und Fehlerzustand-Erkennung	117
6.3.1	Korrelation zwischen Hörempfinden und SOM-Agent-Ausgabe	118
6.3.2	Fazit	123

<b>7 Akustisches Diagnose-System</b>	<b>125</b>
7.1 Motivation, Anforderungen und Umsetzung . . . . .	125
7.2 iMAD - Geräuscherfassung mit dem Diagnosetester . . . . .	125
7.3 mADS - mobiles akustisches Diagnosesystem . . . . .	127
<b>8 Zusammenfassung &amp; Ausblick</b>	<b>131</b>
<b>Literaturverzeichnis</b>	<b>135</b>
<b>Anlagenverzeichnis</b>	<b>149</b>
A.1 Merkmalreduktion und Fingerabdruckberechnung . . . . .	149
A.2 Akustikdiagnose-System . . . . .	165
<b>Selbstständigkeitserklärung</b>	<b>169</b>
<b>Thesen der Dissertation</b>	<b>171</b>
<b>Danksagung</b>	<b>175</b>
<b>Index</b>	<b>177</b>



# Symbolverzeichnis

$\alpha_{SRf}$	Spectral Rolloff Parameter
$\delta_{xx}$	Varianz der Variable $x$
$\delta_{xy}$	Kovarianz der Variablen $x$ und $y$
$acc$	Klassifikationsrate, Erkennungsrate (engl. <i>accuracy</i> )
$AI$	<i>Articulation Index</i>
$C$	<i>Crest Factor</i>
$c_x$	<i>Cepstrum</i>
$dB$	Dezibel
$E$	Erwartungswert
$f(n)$	Bandmittenfrequenz des n-ten Bandes
$fk$	Fehlerklasse, Klasse
$FR$	<i>F-Ratio</i>
$FT$	<i>Fourier Transformation</i>
$FT^{-1}$	<i>inverse Fourier Transformation</i>
$H$	Entropie
$I$	Informationsgehalt
$IG$	Informationsgewinn, <i>Information Gain</i>
$M$	Spektralwerte
$m_{in}$	Merkmalvektor mit redundanten und irrelevanten Merkmalen
$M_N$	normalisierte Spektralwerte
$m_{out}$	reduzierter Merkmalvektor, akustischer Fingerabdruck
$MI$	Transinformation (engl. <i>Mutual Information</i> )

$P$	Wahrscheinlichkeit
$R$	Redundanz
$r$	Korrelationskoeffizient
$RMS$	<i>Root-Mean-Square</i>
$rms$	<i>Root-Mean-Square</i>
$SC$	<i>Spectral Centroid</i>
$SF(t)$	<i>Spectral Fluctuation</i>
$SFI$	<i>Spectral Flatness</i>
$SPL$	Sound Pressure Level, Schalldruckpegel
$SRf$	<i>Spectral Rolloff</i>
$SU$	Symmetrische Unbestimmtheit
$t(n)$	Signalauschnittlänge
$x(t)$	Signal $x$ in abh. von Zeit $t$
$x_{eff}$	Effektivwert des Signals $x$
$x_{max}$	Maximalwert des Signals $x$
$ZCR$	<i>Zero Crossing Rate</i>
$ANN$	<i>Artificial Neural Network</i>
C4.5	C4.5-Merkmalreduktionsverfahren
Cart	<i>Classification and Regression trees</i>
CWT	<i>Continuous Wavelet Transform</i>
dB	Dezibel
DCT	<i>Discrete Cosine Transform</i>
DFT	<i>Discrete Fourier Transform</i>
DVE	<i>Digital Voice Enhancement</i>
DWT	<i>Discrete Wavelet Transform</i>
ETA	<i>Event Tree Analysis</i>
FD	<i>Fisher's Discriminante Analysis</i>

FFT	<i>Fast Fourier Transform</i>
FMEA	<i>Failure Mode and Effects Analysis</i>
FTA	<i>Failure Tree Analysis</i>
FWT	<i>Fast Wavelet Transform</i>
HMM	<i>Hidden Markov Model</i>
IAD	Intelligente Akustikdiagnose
ICE	<i>Inter City Express</i>
ID3	ID3-Merkmalreduktionsverfahren
k-NN	<i>k Nearest Neighbor</i>
KNN	Künstliche Neuronale Netze
LMS	<i>Least Mean Squares</i>
LVQ	<i>Learning Vector Quantization</i>
M	Merkmale
MFCC	<i>Mel Frequency Cepstral Coefficients</i>
ML	<i>Maximum Likelihood</i>
MLP	<i>Multi Layer Perceptron</i>
NB	<i>Naïve Bayes</i>
PCA	<i>Principal Component Analysis</i>
SNR	<i>Signal to Noise Ratio</i>
SOM	<i>Self Organizing Map</i>
SPL	<i>Sound Pressure Level</i>
SQL	<i>Structured Query Language</i>
SVM	<i>Support Vector Machine</i>





# Tabellenverzeichnis

3.1	Konfusionsmatrix für einen Zweiklassenfall . . . . .	19
4.1	Vergleich der Erkennungsraten und Rechenzeiten, aus [Ruf09] . . . . .	57
5.1	Übersicht Fahrzeugmessdaten VW Polo 1.4l . . . . .	60
5.2	Übersicht berechneter Merkmale [Dimension] . . . . .	61
5.3	Variation der Signalauschnittlänge $t(n)$ und resultierende Datensatzlänge $N(t(n))$ . . . . .	63
5.4	Übersicht über die Testdatensätze der <i>UCI Machine Learning Repository</i> . . . . .	65
5.5	Klassifikationsraten $acc$ in % (gemittelt über 25 Durchläufe) für die <i>UCI</i> Datensätze und die mittlere erreichte Klassifikationsrate für das jeweilige Mustererkennungsverfahren über alle Datensätze . . . . .	66
5.6	Bewertung der <i>Fisher</i> -Diskriminanten-Trennbarkeit durch Einteilung in drei Gruppen . . . . .	67
5.7	Trennbarkeit durch FD bei ausschließlicher Verwendung der CAN-Merkmale (Mittelwert, variable Zeitkonstante, aus [Kot08]) . . . . .	68
5.8	Güte des MLP-NN Mustererkenners für ausgewählte <i>Fisher</i> -Diskriminante-Merkmale (FD) . . . . .	69
5.9	Ermittlung des maximalen Koeffizienten von jedem Merkmal $m_1$ zu Merkmal $m_2$ . . . . .	73
5.10	Auflistung der <i>Fisher</i> -Diskriminante-Merkmale mit Korrelationskoeffizient-Rangordnung . . . . .	74
5.11	Auflistung der <i>Fisher</i> -Diskriminante-Merkmale mit <i>Relief</i> -Rangordnung . . . . .	75
5.12	Auflistung der <i>Fisher</i> -Diskriminante-Merkmale mit Transinformations-Rangordnung . . . . .	77
5.13	Auflistung der <i>Fisher</i> -Diskriminante-Merkmale mit Informationsgewinn-Rangordnung . . . . .	78
5.14	Auflistung der <i>Fisher</i> -Diskriminante-Merkmale mit FCBF-Rangordnung . . . . .	80
5.15	Benötigte Rechenzeit der Merkmalreduktionsverfahren im Vergleich . . . . .	82
5.16	Zustandsdefinitionen für einen Datensatz mit zwei verschiedenen Fahrzeugen . . . . .	82
5.17	Exemplarischer Vergleich für Klasse 7, von 13 Merkmalen der Fahrzeuge Polo ( $p$ ) und Scirocco ( $s$ ) für die statistischen Kenngrößen Mittelwert ( $mean$ ), Minimum ( $min$ ) und Maximum ( $max$ ) . . . . .	83
5.18	Erkennungsraten der Datensätze bei 9 Merkmalen über fünf Durchläufe gemittelt . . . . .	84
5.20	Untersuchung zur Allgemeingültigkeit der akustischen Fingerabdrücke . . . . .	84
5.19	Die relevanten Merkmale - akustischen Fingerabdrücke - für die drei generierten Datensätze wurden mit dem <i>Relief</i> -Algorithmus ermittelt. . . . .	85
6.1	Rangordnung anhand der Parameter-/ Klassen-Korrelation . . . . .	93
6.2	Mittlere Erkennungsraten aus [TOB09] . . . . .	94
6.3	Die 6 wichtigsten Merkmale aus dem <i>SEL</i> Datensatz, berechnet mit dem <i>Relief</i> -Algorithmus . . . . .	102

6.4	Auftrittswahrscheinlichkeiten des 2D-SOM-Fehleragenten für die 3 Zustände (C1= fehlerbehaftet, C2= fehlerfrei, U= Unbekannt) bei Variation der verwendeten Merkmale	102
6.5	Auftrittswahrscheinlichkeiten des 3D-SOM-Fehleragenten für die 3 Zustände (C1= fehlerbehaftet, C2= fehlerfrei, U= Unbekannt) bei Variation der verwendeten Merkmale	105
6.6	Die 6 wichtigsten <i>Relief</i> -Merkmale aus dem <i>SEL</i> Datensatz und die dazugehörigen Abtastraten . . . . .	112
6.7	Mittlere Entscheidungsraten des 3D-SOM-Fehleragenten für 2 Algorithmusvarianten	115
6.8	Die mittleren Entscheidungsraten des 3D-SOM-Fehleragenten für 129 Mikrofonaufnahmen . . . . .	117
6.9	Entscheidungsraten des 3D-SOM-Fehleragenten für die Hörbarkeit 0 (724 Innenraumaufnahmen) . . . . .	120
6.10	Entscheidungsraten des 3D-SOM-Fehleragenten für die Hörbarkeit 1 (116 Innenraumaufnahmen) . . . . .	121
6.11	Entscheidungsraten des 3D-SOM-Fehleragenten für die Hörbarkeit 0.1 (227 Innenraumaufnahmen) . . . . .	123
A.1	Wahrheitsmatrix für den Datensatz <i>ALL</i> (a), <i>PSY</i> (b), <i>SPEC</i> (c), <i>SEL</i> (d), <i>UNI</i> (e), <i>ALL-SPEC</i> (f) bei $t(n) = 250ms$ aus Abschnitt 5.3.2 . . . . .	153
A.2	Geordnete Merkmale für die <i>FD</i> -Berechnung aus Abschnitt 5.3.2 . . . . .	154
A.3	Geordnete Merkmale für die <i>FD</i> -Berechnung und anschließender Rangordnung durch das Korrelationskoeffizientverfahren . . . . .	156
A.4	Geordnete Merkmale für die <i>FD</i> -Berechnung und anschließender Rangordnung durch das Mehrklassen-Relief-Verfahren . . . . .	158
A.5	Geordnete Merkmale für die <i>FD</i> -Berechnung und anschließender Rangordnung durch die Transinformation . . . . .	160
A.6	Geordnete Merkmale für die <i>FD</i> -Berechnung und anschließender Rangordnung durch den Informationsgewinn . . . . .	162
A.7	Geordnete Merkmale für die <i>FD</i> -Berechnung und anschließender Rangordnung durch das FCBF-Merkmalextraktionsverfahren . . . . .	164

# Abbildungsverzeichnis

3.1	Konzept zum Aufbau eines akustischen Prüfsystems für den Fahrzeuginnenraum . .	16
3.2	Beschreibung der Datenbankstruktur . . . . .	16
3.3	Vereinfachter Ablauf einer Klassifikationsaufgabe . . . . .	18
4.1	Verfahren zur Merkmalreduktion . . . . .	27
4.2	Alle möglichen Merkmalsätze $m_{in1}$ bis $m_{inX}$ werden durch den Wrapper in Kombination mit dem Klassifikator getestet. . . . .	29
4.3	Vergleich des <i>FCBF</i> -Algorithmus mit anderen Reduktionsverfahren, aus [YLG04]. .	36
4.4	Beschreibungsformen von Klassifikationsverfahren . . . . .	37
4.5	Prinzip der kaskadierten Klassifikation des Merkmalvektors $m$ . . . . .	38
4.6	Die Generierung eines Trainings- und Testdatensatzes anhand der <i>hold-out</i> Methode	39
4.7	Kreuzvalidierungsverfahren am Beispiel eines Datensatzes mit 3 Fehlerklassen und 300 Mustern . . . . .	40
4.8	Merkmalraum; Muster gleicher Klasse sind farblich markiert (grau = $fk_1$ ; schwarz = $fk_2$ ) . . . . .	41
4.9	a) 2-dimensionales Kohonen-Karten-Modell mit 9 Neuronen b) 3-dimensionale Anordnung der Neuronen . . . . .	46
4.10	a) Die Merkmale $x_1$ und $x_2$ des zufällig aus dem Datensatz ausgewählten Musters werden der SOM über die Eingabeknoten präsentiert. b) Distanzmaß: Euklidische Distanz - entspricht dem Satz des Pythagoras bei $m = 2$ . . . . .	47
4.11	Problematik des Adaptionsvorgangs nach Gleichung 4.33 . . . . .	48
4.12	Beispiele zur Berechnung der Distanz vom Gewinnerneuron (schwarz) zu den Nachbarneuronen:	
	a) Eine Linien-Topologie ( $\mathbb{R}^1$ )	
	b) Eine Netz-Topologie ( $\mathbb{R}^2$ )	
	c) Eine Netz-Topologie, jedoch mit 7x7 Neuronen (rote Punkte, Gewinnerneuron ist $w_k(4,4)$ ) . . . . .	49
4.13	Einfluss der Scheitelbreite der Nachbarfunktion . . . . .	50
4.14	Die Zuordnung der Klasseninformation zu den Neuronen erfolgt durch eine Zustands- / Klassendarstellung. . . . .	51
4.15	Schematische Darstellung eines 12x12 Neuronennetz durch eine Komponentendarstellung . . . . .	51
4.16	Aufbau der Verzögerungsstufe; $P$ Verzögerungsgliedern für jedes Merkmal des Eingabevektors $\vec{x}$ . . . . .	54

4.17 Eindimensionale Verteilungsdichtefunktion des Merkmals $m_1$ für zwei Fehlerklassen $fk_1$ und $fk_2$ . . . . .	55
5.1 Vergleich verschiedener Mustererkennungsverfahren auf der Basis von UCI Datensätzen . . . . .	66
5.2 Klassifikationsraten bei Merkmalreduktion durch die <i>Fisher</i> -Diskriminante bei unterschiedlicher Zeitkonstante $t(n)$ . . . . .	70
5.3 Absoluter Korrelationskoeffizient (Farbskala) für den reduzierten Merkmalsatz ( <i>subset</i> )	72
5.4 Vergleich der Korrelationskoeffizient-Klassifikationsraten . . . . .	74
5.5 Vergleich der Relief-Klassifikationsraten . . . . .	76
5.6 Vergleich der Transinformations-Klassifikationsraten . . . . .	77
5.7 Vergleich der Informationsgewinn-Klassifikationsraten . . . . .	79
5.8 Vergleich der FCBF-Klassifikationsraten . . . . .	80
5.9 Merkmalreduktionsverfahren im Vergleich . . . . .	81
6.1 Darstellung der Geräuschunterscheidung anhand des Korrelationskoeffizienten (Quelle: [TOB09]) . . . . .	88
6.2 Zustandsspezifische Korrelationskennlinien (Quelle: [TOB09]) . . . . .	90
6.3 Beispiel SOM-Adaption: links: Gewichte der Neuronen (zwei Klassen: C1 (Sternchen) und C2 (Diamanten)) rechts: Zweidimensionale Neuronenkarte mit Klassenzuordnung (markierte Neuronen)	95
6.4 Komponentendarstellung (rot entspricht einer starken und blau einer geringen Ausprägung) . . . . .	96
6.5 Bewertung eines Zeitsignals durch eine Kohonen-Karte . . . . .	101
6.6 Berechnete Auftrittswahrscheinlichkeiten des 2D-SOM-Fehleragenten bei Variation von $M$ . . . . .	104
6.7 Berechnete Auftrittswahrscheinlichkeiten des 3D-SOM-Fehleragenten bei Variation von $M$ . . . . .	106
6.8 a) Quantisierungsfehler bei verschiedenen Epochenlängen des Trainingsprozesses b) Varianz aus der absoluten Häufigkeit der Gewinnerneuronen für den Trainingsdatensatz . . . . .	107
6.9 Das Histogramm für 600 Trainingsepochen zeigt im Vergleich zu 15000 Epochen eine deutlich schlechtere Verteilung. Dementsprechend ist die Varianz für $T=600$ größer als für $T=15000$ . . . . .	108
6.10 Variation von $\sigma_I$ im Bereich $[0,5...3]$ . . . . .	109
6.11 Variation von $\sigma_F$ im Bereich $[0.0001 \dots 0.7]$ . . . . .	110
6.12 Variation von $\mu_I$ im Bereich $[0.4 \dots 1]$ . . . . .	110
6.13 Variation von $\mu_F$ im Bereich $[0.0001 \dots 0.7]$ . . . . .	111
6.14 Schema zur Interpretation der der Berechnungsschrittweite $SOM\_STEP$ und der Puffergröße $T$ . . . . .	113
6.15 Analyse des Einflusses der Puffergröße von $T$ auf die Erkennungsraten des SOM-Agents . . . . .	113
6.16 Auswertung der Erkennungsraten des SOM-Agents bei variierendem $SOM\_STEP$ .	114

6.17 Analyse des Quantisierungsfehlers bei Variation der Netzstruktur und Neuronenanzahl	116
6.18 Verlauf des Anteils markierter Neuronen bei zunehmender Neuronenanzahl . . . . .	116
6.19 Schema zur Zustandsentscheidung durch den 3D-SOM-Fehleragenten . . . . .	119
6.20 Verlauf der Zustandsentscheidung für den Datensatz 9700 (Bild a) und 9750 (Bild b)	122
7.1 VAS 5052a Diagnosetester mit iMAD-Software (links) und mehrkanalige Mikrofon- einheit (rechts) . . . . .	126
7.2 mADS-Hardware . . . . .	127
7.3 Die Software des mADS bietet diverse Funktionen zur Signalanalyse, Protokollerfas- sung, Wiedergabe und Berechnung eigener Algorithmen. . . . .	128
7.4 Aufruf der Matlab®-DLL aus der mADS-Software . . . . .	129
A.1 Strukturierung der Datenablage in der entwickelten SQL-Datenbank. . . . .	149
A.2 Häufigkeitsverteilungen der Merkmalsätze/ Muster . . . . .	150
A.3 Gleichverteilung der Merkmalsätze/ Muster . . . . .	151
A.4 Fisher-Diskriminanten Matrix für den Datensatz <i>ALL</i> , <i>PSY</i> , <i>SPEC</i> , <i>SEL</i> , <i>UNI</i> und <i>ALL-SPEC</i> bei $t(n) = 250ms$ aus Abschnitt 5.3.2 (auf der Farbskala ist der FD-Wert $f$ abgebildet) . . . . .	152
A.5 Extrahierter absoluter Korrelationskoeffizient mit $r < 0.3$ . . . . .	155
A.6 Amplitudenfrequenzgang der Mikrofonkapsel des Typs DPA 4060. Es sind die Fre- quenzgänge von 4 Kapseln des gleichen Typs abgebildet. . . . .	165
A.7 Amplitudenfrequenzgang der Mikrofonkapsel des Typs MEO-65PN-02-623. Es sind die Frequenzgänge von 4 Kapseln des gleichen Typs abgebildet. . . . .	166
A.8 Richtcharakteristik bei 0.2 kHz und 1 kHz für DPA 4060 (linke Seite) und MEO-65PN (rechte Seite) . . . . .	167
A.9 Richtcharakteristik bei 6.3 kHz und 12.5 kHz für DPA 4060 (linke Seite) und MEO- 65PN (rechte Seite) . . . . .	168



# 1 Einleitung

Das Innenraumgeräusch eines Fahrzeugs beeinflusst maßgebend das Qualitätsempfinden des Kunden. In Zusammenarbeit mit der Konzernforschung der Volkswagen Aktiengesellschaft wird im Rahmen dieser Arbeit das Gebiet der akustischen Diagnose für die Produktnutzungsphase bearbeitet. Durch Alterung und Verschleiß verändert sich das Innenraumgeräusch und wird bis zu einem gewissen Punkt als normales Fahrzeuggeräusch akzeptiert. Sobald jedoch Geräusche auftreten, die vom Kunden nicht als normal eingestuft werden, liegt die Aufgabe der Bestimmung der Fehlerursache bei den Werkstätten. Durch eine schnelle und sichere Fehlerlokalisierung und -behebung kann die Kundenzufriedenheit verbessert und die Kosten für den Kunden und den Hersteller (zum Beispiel bei bestehenden Garantieansprüchen) maßgebend reduziert werden. Dabei verhält es sich jedoch so, dass die Interpretation der auftretenden Geräusche und dessen Zuordnung zu einem Bauteil allenfalls nur von Experten schnell und zuverlässig vorgenommen werden kann. Aufgrund der Erfahrung und des Wissens werden typische Geräusche auch unabhängig vom Fahrzeugtyp wiedererkannt (zum Beispiel defekte Radlager oder Gelenkwellen im Antriebsstrang). Damit gehört die akustische Beurteilung im Werkstattbereich zu einem Bereich, der bisher nicht automatisiert oder durch geführte Fehlersuche (in Verbindung mit einem Werkstatt-Fahrzeugdiagnosegerät) bearbeitet wird. Aus diesem Grund stellt das Forschungsthema "automatisierte akustische Fahrzeuginnenraumdiagnose" ein neues Forschungsgebiet dar. Dafür soll ein Modell zur automatischen Analyse von Geräuschen entwickelt und an Versuchsfahrzeugen mit Fehlerbauteilen getestet werden.

Es bestehen diverse Untersuchungen und Verfahren, die eine akustische Qualitätssicherung zulassen. So werden beispielsweise in [OM09] Motoren anhand ihrer emittierten Schwingungen hinsichtlich auftretender Fehler analysiert. Grundlage für die Erkennung von Schadensmustern ist die Aufnahme von Sensorsignalen. Meist werden Beschleunigungssensoren oder Mikrofone verwendet. Durch moderne Signalverarbeitung werden die relevanten Muster aus Rauschen, Hintergrundgeräuschen und anderen dem Sensorsignal überlagerten Anteilen extrahiert. Dafür werden die Signale in mathematische Beschreibungsformen (zum Beispiel spektrale oder Wavelet-Analysen) transformiert, welche auch als Merkmale bezeichnet werden. Durch Verfahren zur Mustererkennung, wie sie beispielsweise bei der Erkennung von Musiktiteln ([Roh03, MSS02]) eingesetzt werden, können die Eigenschaften der Merkmale interpretiert und mit dem Zustand (beispielsweise der Musikrichtung oder des Interpreten) verknüpft werden. Das Modell zur Mustererkennung entspricht somit einem Wissensspeicher, welcher anhand einer gelernten Erfahrung Entscheidungen trifft. Dabei stellen die Merkmale den Schlüssel zum Erfolg dar, da diese eindeutig relevant für die zu erkennenden Zustände sein müssen; sie bilden einen akustischen Fingerabdruck. Das herausfordernde Ziel, ein System zur automatischen Bewertung des Innenraumgeräuschs zu entwickeln, wird maßgeblich durch die komplexe Innenraumakustik und die Identifikation der zur Fehlererken-

nung notwendigen akustischen Fingerabdrücke (für den Serienklang und die auffälligen Geräusche) beeinflusst. Dafür müssen die relevanten Kenngrößen zur Beschreibung der Akustik wiederum anhand geeigneter Verfahren identifiziert werden.

Neben der Unterstützung der Werkstätten durch eine geführte Analyse des Innenraumgeräuschs stellt eine weitere Vision die kontinuierliche Analyse des Fahrzeuginnenraumgeräuschs (zum Beispiel durch ein Akustikdiagnose-Steuergerät in Verbindung mit dem Freisprechmikrofon) dar. Eine Symbiose mit bestehender Hardware wie DSP-Verstärker, Freisprecheinrichtungen oder ähnlichen bestehenden Signalverarbeitungseinheiten ist dabei denkbar. Das Steuergerät berechnet fortwährend akustische Fingerabdrücke und vergleicht den Soll-Zustand mit dem Ist-Zustand. Bei einer Abweichung wird ein Fehlerspeichereintrag erzeugt. Dieser wird in den Werkstätten ausgelesen und mit einer Datenbank verglichen. Es ist ersichtlich, dass dieses Szenario eine interdisziplinäre Herausforderung darstellt, welche eine Vernetzung von Werkstätten, technischem Servicecenter/Qualitätssicherung und Produktion bedingt. Da diese Industriepromotion direkt bei der Volkswagen AG entstand, konnten neben den Untersuchungen von Ansätzen zur Machbarkeit, der Algorithmenentwicklung und dem exemplarischen Test an Versuchsfahrzeugen auch die Vernetzung und Etablierung grundlegender Infrastruktur (beispielsweise eine Konzern-Datenbank zur Erfassung von Kunden-Fahrzeuggeräuschen) vorangetrieben und unterstützt werden.



## 2 Stand der Technik

Ein Überblick über Techniken und Anwendungen zur akustischen Diagnose ist in diesem Kapitel gegeben. Dabei werden diverse Anwendungen in Bezug zum Automobil vorgestellt und bewertet. Im Folgenden wird auf akustische Analyseverfahren in der Automobilindustrie, Spracherkennung, *Product Lifecycle Management* (PLM), akustische Analysen zur Qualitätssicherung und auf die Generierung akustischer Merkmale und deren Interpretation eingegangen.

### 2.1 Akustische Kamera, Sprechererkennung & Spracherkennung, Sprachverbesserung

Derzeit werden sowohl im Entwicklungsprozess als auch während der Nutzung eines Fahrzeugs diverse akustische Verfahren zur Analyse von Schallwellen angewendet. Aus diesem Grund werden in diesem Abschnitt verschiedene Anwendung im Bereich Fahrzeugentwicklung und -nutzung, die derzeit im Einsatz sind, genannt und erläutert. Die *Akustische Kamera* (*Acoustic Camera*<sup>1</sup>, [Ker08]) gehört im Bereich der Fahrzeugentwicklung zu einer der besten akustischen Analysemethoden. Dieses System, welches aus mehreren Mikrofonen, die in einem Array angeordnet sind, und einer Aufnahmeeinheit mit Signalverarbeitung (z.B. *Beamforming*) besteht, ermöglicht eine Abbildung des Schallpegels auf einem Videobild. Es ermöglicht eine akustische Diagnose hinsichtlich der Lokalisierung der Fehlerquelle. Aufgrund der Größe (diese variiert je nach Array-Geometrie), Preis und der Signalverarbeitung ist die *Akustische Kamera* größtenteils nicht für den Einsatz in Werkstätten geeignet. Darüber hinaus liefert diese keine Aussage über das Fehlerbauteil, sondern detektiert lediglich dessen Hauptschallkomponente. Dadurch ist dieses System dennoch sehr gut für den Entwicklungsprozess geeignet, um zum Beispiel Optimierungen hinsichtlich Gehäusesteifigkeiten und damit verbundenen Schallemissionen an Bauteilen vorzunehmen. In [Ker08] sind Methoden für eine verbesserte Signalverarbeitung vorgestellt, aber die Eignung für Innenraummessungen konnte bisher nur bedingt gezeigt werden. Eine weitere Alternative im Bereich der Lokalisierung ist zum Beispiel in [Ber08] genannt, die Blinde Quellentrennung (auch *Blind-Source-Separation* genannt, BSS) zur Ortung. Dabei wird der Ansatz verfolgt, dass Fehlergeräusche abhängig vom und der Klangcharakteristik voneinander getrennt werden können und dadurch anhand der ermittelten unterschiedlichen Verzögerungszeiten in den Transferpfaden ein Rückschluss auf den Entstehungsort möglich ist. In der Arbeit wurde ein Modell entwickelt, was die Untersuchung der verschiedenen BSS-Algorithmen ermöglicht. In Bezug auf reale Geräusche wurden jedoch nur mittelmäßige bis schlechte Ergebnisse erreicht.

---

<sup>1</sup><http://www.acoustic-camera.com/>

Abgesehen von den Verfahren zur Lokalisierung der Schallquelle werden zum Beispiel auch Systeme zur Sprechererkennung und Spracherkennung im Fahrzeug verwendet. Bei dem im VW Multivan eingesetzten DVE-System (*Digital-Voice-Enhancement*, [Ste04]) wird der aktive Sprecher erkannt und das über ein Mikrofon aufgenommene Sprachsignal verstärkt auf den anderen Sitzplätzen wiedergegeben. Dadurch wird die Verständlichkeit bei Unterhaltungen im Fahrzeug verbessert. Im Bereich Infotainment werden Spracherkenner ([ST01, RJ93, PK08]) für Funktionen eingesetzt, wo ein begrenzter Wortschatz benötigt wird. Prinzipiell wird zwischen

- Sprecher-unabhängiger Spracherkennung,
- Sprecher-abhängiger Spracherkennung

unterschieden. Bei der Sprecher-unabhängigen Spracherkennung kann das System ohne vorheriger Trainingsphase benutzt werden; meist wird dafür der Wortschatz jedoch begrenzt. Bei Sprecher-abhängigen Spracherkennern muss das System auf die jeweilige Aussprache des Benutzers trainiert werden, dafür kann jedoch ein größerer Wortschatz verwendet werden. Die Spracherkenner lassen sich verallgemeinernd in mehrere Bestandteile zerlegen. In der Vorverarbeitungsphase werden die analogen Mikrofonsignale digitalisiert und gefiltert, um Sprache von Hintergeräuschen zu trennen. Anschließend werden die Sprachsignale transformiert, um Merkmale zu generieren. Eine Transformation ist z.B. die Berechnung der FFT zum Verarbeiten einzelnen Frequenzanteile. Beispielsweise wird das Cepstrum [YHC06] als Merkmal verwendet. Dabei wird erneut eine FFT aus dem logarithmischen Amplitudenspektrum berechnet um Periodizitäten im Spektrum erkennen zu können. Die Merkmale werden anschließend in einem Algorithmus zur Spracherkennung verarbeitet. Hier kommen häufig *Hidden-Markov-Models* (HMM, [RJ93, Fin03, Rab90]) zum Einsatz. Alternativ werden in [WHH<sup>+</sup>89] Neuronale Netze verwendet. Eine hybride Methode wird in [ZWZ07] vorgestellt. Dabei wird ein HMM-Modell mit einem genetischen Algorithmus [Gol89] optimiert, um die Erkennungsraten weiter zu steigern. Im Automobil werden Spracherkenner vorwiegend in Sprachdialogsystemen in Verbindung mit der Infotainment- und Multimediasystembedienung verwendet. Die Medizintechnik stellt ein weiteres Anwendungsgebiet von Spracherkennungsverfahren dar. Es ist ersichtlich, dass die Spracherkenner in verschiedenen Bereichen angewendet werden, dabei jedoch nur ein begrenzter trainierter Wortschatz zur Verfügung steht. Die Methodik zur Erkennung der Sprache könnte jedoch in Bezug auf die angestrebten Ziele dieser Arbeit, für eine automatische Erkennung und Unterscheidung der Fahrzeuggeräusche angewendet werden.

## 2.2 Product Life-cycle Management (PLM)

Die Planung eines Produktlebenszyklus' durch ein strategisches Konzept [Sen09] wird als PLM bezeichnet. Nach Entwurf, Konstruktion und Produktion folgt der wichtige Bereich Dienstleistung, welcher z.B. durch Datenverarbeitung und -beobachtung, Analyseprozesse und Serviceeinrichtungen abgedeckt wird. Bezogen auf die geplante akustische Diagnose während der Kundennutzung, gibt es Untersuchungen, die sich mit der Verarbeitung von Felddaten in der Produktnutzungsphase auseinandersetzen. In [AFHN08] wird die Analyse der Produktnutzungsphase fokussiert, im Gegensatz zu der weit verbreiteten Unterstützung (durch Dienstleistungen wie Wartung und Instandhaltung oder Kundenservice) im Bereich der frühen Produktlebensphase. Dabei wird ein

Konzept ausgearbeitet, um die Produktnutzungsphase direkt in den PLM-Prozess zu integrieren. Dafür sollen über “*Condition Monitoring*” (CM) Ergebnisse aus der Produktnutzungsphase in die Entwicklung der Folgegeneration eines Produktes einfließen. Aus der Zustandsüberwachung geht hervor, dass vorwiegend Diagnose zur Prävention von Stillständen und Diagnose von Schäden eingesetzt wird. Dabei werden Kupplungen, Pumpen, Getriebe aber auch Wafer-Produktion<sup>2</sup>, Gießverfahren, Werkzeugmaschinen, Motoren und Fahrzeuge überwacht. Es geht aus [AFHN08] hervor, dass neue “Dienste und Serviceleistungen wie Verfügbarkeit und Zuverlässigkeit, Ferndiagnose, Ersatzteilversorgung, oder zustandsorientierte Instandhaltung” neue Geschäftsfelder ermöglichen, welche derzeit bestenfalls rudimentär unterstützt werden. Dabei “kommen als bekannte Verfahren die Fehlermöglichkeits- und Einflussanalyse (FMEA), die Ereignisbaumanalyse (ETA), die Fehlerbaumanalyse (FTA) und Markov Prozesse sowie als neue Verfahren *Computational Intelligence* Anwendungen als Gebiet der Künstlichen Intelligenz oder graphbasierte Verfahren zum Einsatz”. In [Edl01] wird ebenfalls “ein Ansatz vorgestellt, der es ermöglicht, auf Basis von Felddaten Teilaufgaben der Produktentwicklung und des Services zu optimieren” (Zitat).

## 2.3 Akustische Analysen zur Qualitätssicherung

Bezogen auf die Potentiale des PLM-Prozesses nutzt der Technische Service der Qualitätssicherung bei Volkswagen das Geräuschlexikon als Diagnosehilfe von Beanstandungen [Put09]. Derzeit werden 40 Geräuscharten mit insgesamt ca. 150 Geräuschbeispielen zur Analyse von Fehlergeräuschen verwendet. Dieser Ansatz fördert bisher lediglich eine einheitliche Kommunikation über die Geräuscharten, unterstützt jedoch nicht die Werkstätten beim schnelleren Beheben von Fehlern.

Akustische Untersuchungen mit definierten Randbedingungen zur Erkennung von Produktfehlern finden größtenteils im Bereich der Produktion und Qualitätssicherung statt. In Zusammenarbeit mit der Fa. Bosch wurden von [Jah01, Ste99, Wir00] verschiedene Dissertationen im Bereich der akustischen Qualitätssicherung erstellt.

In [Ste99] wird die Analyse und Auswahl von Verfahren zur akustischen Güteprüfung in der Serienfertigung vorgestellt. Dabei werden die Untersuchungen auf Produkte aus der Kraftfahrzeugausrüstung und Automatisierungstechnik ausgerichtet. Wie in Abschnitt 2.1 bereits erwähnt, wird auch hier der Fokus auf die Merkmalauswahl gelegt. Als Resultat soll ein System zur objektiven Güteprüfung umgesetzt werden. Im Bereich der Merkmalsextraktion werden Psychoakustische Verfahren (dazu zählt zum Beispiel die Lautheit, Schärfe und Rauigkeit) und Zeit-Frequenz-Verteilungen wie Spektral- und Wavelet-Analyse betrachtet. Anschließend folgt die Merkmalsbewertung durch Fisher-Diskriminante, F-Ratio und Transformation des Merkmalraums. Bei den Methoden der Merkmalsklassifikation werden geometrische und statistische Verfahren, Fuzzy- und Neuronale-Systeme vorgestellt. Die objektive Güteprüfung wurde für drei industrielle Anwendungen erläutert. Im ersten Schritt wird ein Prüfkonzept für Dieseleinspritzkomponenten vorgestellt. Dabei wird als Merkmal

---

<sup>2</sup>Ein Wafer stellt in der Halbleiterindustrie die Grundplatte (auch Substrat genannt) dar, auf dem verschiedene elektronische Bauelemente hergestellt werden.

(über der Zeit) die “Diskrete Wavelet-Transformierte der Beschleunigung” verwendet. Die Beschleunigung wird dabei über am Prüfstand montierte Beschleunigungsaufnehmer erfasst. Über die “Bestimmung der zeitl. Differenz des Maximums dieser Größe zum Maximum einer zweiten physikalischen Größe (Druck in der Pumpe)” wird eine “Klassifikation durch Schwellenwertklassifikator” erreicht und die Eignung des Verfahrens nachgewiesen. Die nächste beschriebene Anwendung bezieht sich auf “Qualitätsüberwachung von Laserbeschriftungen”. Dabei soll über Luftschall eine Qualitätskontrolle des Schreibvorgangs an einem Produkt der Automobilindustrie realisiert werden. Die Erreichbarkeit der Ziele “Langzeitüberwachung der Beschriftungsqualität” und “Kurzzeitüberwachung der Zeichenvollständigkeit” werden ebenfalls anhand von Versuchen auf Prüfständen untersucht. Die Mikrofone wurden innerhalb der Beschriftungskabine positioniert wobei diese nicht schalldicht ist. Es kam jedoch heraus, dass für eine Prüfung mittels Luftschall eine sorgfältige Kabinenauskleidung notwendig ist. Für die Langzeitüberwachung werden die Mikrofonsignale bandpassgefiltert, der RMS<sup>3</sup>-Wert berechnet und über mehrere Messungen gemittelt. Über einen Langzeit-Schwellenwert wird eine Abweichung vom Soll-Zustand erkannt. Bei der Kurzzeitüberwachung wird für jedes bandpassgefilterte Mikrofonsignal die Hüllkurve extrahiert und anschließend über einen Schwellenwertvergleich ausgewertet. Aufgrund der prägnanten Ausprägung eines für den Schreibvorgang typischen Signalverlaufs im Bereich von 15kHz konnten sowohl für die Kurzzeitüberwachung als auch für die Langzeitüberwachung die Funktionsfähigkeit nachgewiesen werden.

In [Wir00] werden ebenfalls akustische Verfahren im Rahmen einer Dissertation entwickelt, anhand denen Bauteile untersucht und beurteilt werden können. Die Voruntersuchungen wurden in die Bereiche Signalklassifikation, digitale Signalanalyse und automatische Fehlerdiagnose unterteilt. Sein Konzept des Diagnosesystems ist in zwei Hauptbestandteile eingeteilt. Im ersten Schritt werden aufgenommene Mikrofonsignale über ein Qualitätssicherungsmodell in “gut, schlecht und unsicher” gruppiert und anschließend die schlechten Signale durch ein Fehlerdiagnose-Modell eindeutig klassifiziert. Diese kaskadierte Form der Fehlererkennung hat den Vorteil, dass die schnelle Qualitätskontrolle wenig Rechenaufwand benötigt und dadurch nur fehlerhafte Signale die rechenaufwendigen Fehlerdiagnose durchlaufen. Die Grundlagen und Theorie zur Merkmalgenerierung und -analyse (Korrelation, Filter, Frequenzanalysen, Wavelet) werden sehr umfassend beschrieben und anschließend auf Fensterhebermotoren und Automobilgeneratoren angewendet. Bei den Fensterhebermotoren werden die Geräusche in 4 Gruppen unterteilt:

- fehlerfrei
- Lagerfehler
- Getriebefehler
- Bürstenfehler

Die Analysen werden anhand Luftschall in einer optimierten Messkabine durchgeführt. In Bezug auf die Generatoren hat [Wir00] einen Generator für Prinzipuntersuchungen ausgewählt. Anhand des Körperschallsignals sollen folgende Fehlerarten untersucht werden:

---

<sup>3</sup>Bezeichnet den Effektivwert (*root-mean-square*, rms) und entspricht somit dem quadratischen Mittelwert eines Zeitsignals.

- FB (foreign body): "Ein Fremdkörper (z.B. Kabelabschnitt) ist in das Generatorgehäuse gefallen und erzeugt zwischen Rotor und Stator Geräusche."
- FT (fan to tube): "Die Lüfterschaufeln schlagen an die Papierisolierung der Wicklung."
- FW (fan to wire): "Die Lüfterschaufeln schlagen an die Ständerwicklung."
- RP (rotor to paper): "Die Klauenpole des Läufers streifen an der Papierisolierung zwischen Ständerwicklung und Ständersteg." Weiterhin findet eine Unterteilung in RPL (rotor paper loud) und RPQ (rotor paper quiet) statt.
- EOT (end of tube): "Die Lüfterschaufeln schlagen an das Ende der Papierisolierung der Statorzuleitung."

Für die Untersuchungen wurden aufgrund einer zu geringen Anzahl von Datensätzen und schlechter Differenzierbarkeit die Anzahl der Fehlerklassen reduziert: "Aus den fünf Fehlerklassen werden ... im weiteren nur die drei Fehlerklassen FT, EOT und RPL betrachtet." In den Auswertungen durchlaufen die Testsignale (insgesamt 137 Fensterhebermotoren-Aufnahmen und 233 Generatoren-Aufnahmen) als erstes die Fuzzy-Qualitätskontrolle und anschließend die Fuzzy-Klassifikation. Dabei wird das Signal in Frequenzbänder unterteilt, und die mittlere Leistung jedes Bandes bildet ein Merkmal. Der Frequenzbereich von 20 bis 15.000 Hz ist in 8 Abschnitte, die so genannten Oktaven, unterteilt. Die Einteilung in 8 Oktaven ist für manche Betrachtungen zu grob, weshalb beim Terzspektrum jede Oktave nochmals in drei Bänder unterteilt wird. Anhand der Abweichung der Merkmale zu dem bekannten mittleren Muster von fehlerfreien Fensterhebermotoren wird eine Bewertung und damit Klassifikation vorgenommen. Die Klassifikation der genauen Fehlerursache bei fehlerhaftem Geräusch erfolgt analog durch einen Vergleich des aktuellen Musters mit einem bekannten mittleren Muster der bekannten Fehlerklassen. Zusammengefasst stellt die von [Wir00] beschriebene Umsetzung eine spektrale Klassifikation bei definierten Umgebungsbedingungen dar. Es wurden explizit Fehler aus dem Analyseprozess ausgeschlossen, für die zu wenige Datensätze verfügbar waren.

In [Jah01] soll im Rahmen einer Industriepromotion eine frei programmierbare akustische Fehlerdiagnoseplattform entwickelt werden. Dadurch sollen Ergebnisse im Rahmen vergangener Arbeiten der Fa. Bosch integriert und modulare Anforderungen erfüllt werden. Die wichtigsten sind:

- Einfache Erweiterbarkeit des Funktionsumfangs
- Skalierbare Datenerfassungshardware
- Automatische Merkmalselektion anhand einer Stichprobe (Trainingsmenge)
- Klassifikatoren für die Fehlerdiagnose
- Drifterkennung und -kompensation

Wobei die letzten drei Punkte den wissenschaftlich herausfordernden Teil darstellen. Im Rahmen der Merkmalberechnung wurden Frequenz-, Statistik-, Drehzahl- und Pegel-Analysen, wie in [Wir00, Ste99] beschrieben, genutzt. Über eine Merkmalkorrektur wurden darüber hinausgehend Veränderungen im Frequenzgang bei Verwendung verschiedener Produkttypen korrigiert. Es geht hervor, dass die Änderung der Merkmale bei unterschiedlichen Produkttypen betrachtet werden

müssen, „...da keine typunabhängigen fehlerspezifischen Merkmalausprägungen existieren“ (Zitat). Über Verteilungsmodelle der Merkmale werden charakteristische Ausprägungen definiert und klassifiziert. Drifteffekte werden z.B. durch Temperaturschwankungen, Werkzeugverschleiß oder Schwankungen von Materialparametern hervorgerufen. Diese führen zu einer Verschlechterung der Klassifikationsgüte/ Erkennungsrate<sup>4</sup> und werden durch regelmäßige Überwachung des Mittelwerts und der Streuung berücksichtigt. Zum Teil können diese auch kompensiert werden. Das entstandene Prüfsystem wird unter dem Namen *QuickCheck* [Pol05] von der Firma Polytec vertrieben und beinhaltet, wie aus den Veröffentlichungen von [Jah01, Ste99, Wir00] hervorgeht, folgende Eigenschaften:

- Datenerfassung (Anz. Messkanäle, Bandbreite, Kalibrierung...)
- Signalgenerator (Sinus, Sägezahn, Puls...)
- Datenauswertung (Pegel und *signal to noise ratio* SNR, Merkmalberechnung, Schwellenwertklassifikation...)
- Visualisierung (Spektrum über mehrere Messungen, Merkmale, Merkmalverlauf...)
- Datenarchivierung (Zeitdaten als .wav-Dateien, prüftypspezifische Ablage...)

Dieses System ist also auf definierte akustische Prüfbedingungen, zum Beispiel für Prüfstandsanalysen geeignet. Weiterhin scheint, bezogen auf die Anforderungen einer akustischen Analyse im Fahrzeuginnenraum, die starke Gewichtung auf spektrale Merkmale ungeeignet, da ein sehr dominantes breitbandiges Fahrgeräusch (Motor-, Wind-, Reifengeräusch usw.) eine unvorhersehbare Überlagerung der Fehlergeräusche darstellt.

Dennoch werden ähnliche Systeme mit Erfolg z.B. bei der Motorenfertigung eingesetzt. In [OM09] wird das „delta-ANALYSER“ Messsystem zur Schadensfrüherkennung der Reilhofer KG vorgestellt. Dieses ermöglicht eine geräuschbasierende Prüfstandsanalyse anhand von Körperschall. „Man kann anhand der aufgezeichneten Daten aus heutiger Sicht relativ genau den Schadensort und -zeitpunkt ermitteln, sowie die Schadenentwicklung beobachten“. Über Ordnungs-, Frequenzanalyse und Relativspektrum wird der Zustand des Aggregats beurteilt. Weiterhin werden „Hunting Wavelets“ erwähnt, mit der Aussage „Schläge, Schleif- und Kratzgeräusche, nahezu alle unerwünschten Effekte werden damit erkannt“. Jedoch werden keine weiterführenden Verweise genannt. Es scheint sich jedoch um ein spezielles Wavelet zu handeln, welches höhere Frequenzbereiche sehr gut abbildet. Der Nutzen und die Wirtschaftlichkeit haben sich bei mehrjährigen Einsätzen des Systems gezeigt. Dadurch konnten „Aggregate vor schweren Folgeschäden bewahrt werden und somit Aufschluss zu Fehlerursachen gewonnen werden“. Im Ausblick wird jedoch deutlich, dass das durchaus noch interessantere und auch schwierigere Feld die genaue Lokalisierung bzw. Erkennung der defekten Baugruppe sei. Dafür sollen für die Analysesoftware der Zukunft Schadensbilder im „Versuch mit ihren unterschiedlichen Schadensbildern und deren dazugehörigen Ordnungen“ in die Produktion übertragen und „diese Erkenntnisse für die Mustererkennung in der Produktion herangezogen werden“.

Ein etwas älteres, dennoch interessantes Anwendungsbeispiel eines akustischen Prüfsystems liefert [FWH98]. Schon damals war eine objektive Bewertung der Geräusche des Prüflings - in diesem Fall

---

<sup>4</sup>Beschreibt das Verhältnis der vom System erkannten zu der tatsächlich Klasse (Fehler) für eine definierte Anzahl von Mustern.

Kassettenlaufwerke - gefordert, um "Imageschäden und Folgekosten aufgrund von Kundenreklamationen zu vermeiden". Sehr interessant ist das benutzte Klassifikationsverfahren, welches in den aktuellen Veröffentlichungen wenig Aufmerksamkeit erhielt, ein Neuronales Netz in der Form der selbstorganisierenden Merkmalkarte (*Self-Organizing Map*, SOM) nach Kohonen [Koh01]. Diese selbstorganisierenden (unüberwachte) Algorithmen passen sich selbstständig an die Verteilung der Eingabedaten an, ohne dass näheres Wissen (des Beobachters) einfließen muss. Weitere Informationen zu selbstorganisierenden Karten zum Lernen von diskreten und kontinuierlichen Abläufen sind in [Fle05] gegeben. Wie bei den anderen Verfahren (Schwellenklassifikatoren, Fuzzy, HMM usw.) wird ebenfalls eine Grundmenge an Daten benötigt, um den Prüfalgorithmus zu konditionieren, jedoch ist die selbstorganisierende Karte leicht zu interpretieren. Verallgemeinert kann der Prüfalgorithmus in einen Trainingsprozess - dabei passen sich die Neuronen an die Trainingsdaten an - und einen Markierungsprozess (*labeling*) - wo jedes Neuron eine Klassen- bzw. Fehlerinformation erhält - eingeteilt werden. Es zeichnet sich ab, dass ein Großteil der akustischen Prüfsysteme bei sehr definierten Umgebungsbedingungen eingesetzt werden (z.B. Prüfstandsbetrieb).

Ein Ansatz zur Umsetzung eines mobilen akustischen Prüfsystems wird in [Guo05] vorgestellt. Im Fokus der Dissertation steht die Entwicklung eines Fahrwerk-Monitoringsystems. Ziel "ist die Algorithmenentwicklung für eine On-Board-Diagnose von Schäden an sicherheitsrelevanten Fahrwerkskomponenten sowie eine schnelle Entgleisungsdetektion". Dadurch soll bei der Deutschen Bahn AG langfristig eine "Optimierung von aktuellen Instandhaltungsprozessen durch Erhöhung des Anteils an zustandsorientierten Instandhaltungsmethoden" erreicht werden. Ebenfalls werden zum Teil bereits genannte Veröffentlichungen angeführt, darüber hinaus auch weitere z.B. aus dem Bereich Motorsteuergerätoptimierung von [Sch07, Köh98], Fahrwerkszustanderkennung [Wue97] oder Triebwerksdiagnose [Bau97]. Beschleunigungsdaten am Fahrwerk dienen als Grundlage zur Entwicklung der Algorithmen. Die Messdatengewinnung erfolgt an einem ICE-Mittelwagen. Es wurden Diagnosekonzepte für Radlaufflächen, Radsatzlager, Radsatzführung, Schlingerdämpfer und Entgleisung erarbeitet. Es wurde eine Aufteilung und damit verbundene komponentenbezogene Algorithmenentwicklung umgesetzt, da "die Probleme der Merkmalselektion und der automatischen Regelgenerierung unter Berücksichtigung von z.B. mehreren Hundert Merkmalen noch nicht völlig gelöst" sind. Bei Verwendung von Merkmalselektionsalgorithmen, wie z.B. Vorwärtsselektion mit *Support Vector Machine* (SVM, [Vap95, Vap98]) geht hervor, dass die vorhandenen Ansätze in der Regel eine suboptimale Lösung des Problems liefern. Aufgrund derer hohen Komplexität und daraus resultierender schlechter Interpretierbarkeit stellt die regelbasierte Diagnose ein interessantes Gebiet dar. "Die Regeln können sowohl aus Entscheidungsbäumen, neuronalen Netzen oder der SVM extrahiert, als auch mit Hilfe der Regelinduktion gefunden werden" (Zitat). Die umgesetzten Verfahren zur Detektion basieren auf der Analyse von Grenzwertüberschreitungen, Abstandsschätzung und ähnlichen Verfahren zur Auswertung der Beschleunigungssignale. Wiederum ist ersichtlich, dass auch in diesem Fall wenige Ausreißer in den Messdaten vorliegen und diese Normalverteilt sind. Aus diesem Grund können Teilaufgaben der Diagnose sogar regelbasiert erfolgen.

## 2.4 Prüfsystem-Merkmale im Überblick

Wie in dem vorigen Abschnitt beschrieben, sind in den meisten akustischen Prüfsystemen spektrale Merkmale der Schlüssel zum Erfolg. Wobei diese Merkmale zum Teil aus Beschleunigungs- oder Luftschallsensoren generiert wurden. Dennoch wurde in allen Veröffentlichungen darauf geachtet, dass die Sensoren möglichst nahe dem Prüfling befestigt werden und wenn möglich die Umgebungsgeräusche durch z.B. optimierte Prüfkabinen unterdrückt werden. Durch diese Maßnahmen wurden dann die Sensorsignale in den Frequenzbereich transformiert und Überhöhungen bzw. Auffälligkeiten erkannt.

Das Amplitudenspektrum bezeichnet den Betragsverlauf der Amplitude über die Frequenz eines Signals und bildet damit nur einen kleinen Bereich der Wahrnehmung des menschlichen Gehörs ab. Das menschliche Gehör, verknüpft mit den anderen Sinnesorganen, ermöglicht weitaus komplexere Analysen des Schallfelds. Zum Beispiel ermöglicht selektives Hören (Cocktailparty-Effekt) die Trennung einer Schallquelle aus mehreren überlagerten Schallquellen. Das Gehör adaptiert sich auf eine Schallquelle und erreicht dadurch eine Störschallunterdrückung für die störenden Umgebungssgeräusche. Ziel der Merkmale, welche für ein akustisches Prüfsystem verwendet werden, ist es, die durch das Ohr wahrnehmbaren Geräusche mit mathematischen Beschreibungen abzubilden. Das Amplitudenspektrum ermöglicht z.B. eine Interpretation des Schalldruckpegels über der Frequenz, der Bezug zur Phase, Signalverläufen und der emittierenden Schallquelle(n) geht dabei jedoch verloren. Deshalb wird die Interpretation des in den spektralen Bereich transformierten Sensorsignals gerade für Prüfsysteme, wo das Nutzsignal von überlagernden Geräuschen verdeckt wird, nicht ausreichen. Weitere Merkmale wie z.B. Cepstrum, DCT, Wavelet oder auch psychoakustische Merkmale wurden in den erläuterten Veröffentlichungen des Abschnitt 2.3 genannten und sind im Folgenden aufgelistet:

- Spektrale-Analyse
  - Fourier-Transformation
  - Diskrete-Fouriertransformation (DFT)
  - Schnelle-Fouriertransformation (FFT)
  - Cepstrum
- Wavelet-Analyse
  - kontinuierliche Wavelet-Transformation (CWT)
  - Diskrete Wavelet-Transformation (DWT)
  - Schnelle Wavelet-Transformation (FWT)
- Statistik-Analyse
  - Extremwert
  - Varianz
  - Krestfaktor



- Effektivwert (RMS)
- Verteilung
- Korrelation
- Kurtosis
- Schiefe
- Standardabweichung
- Crestfaktor
- Psychoakustik-Analyse
  - Lautheit
  - Schärfe
  - Rauigkeit
  - Schwankungsstärke

Die Psychoakustik beschreibt mit ihren gehörgerechten Parametern den Zusammenhang zwischen der Empfindung des Schalls und den physikalischen Schallfeldgrößen wie Pegel, Frequenz oder Modulationsgrad. Sie stellt also für alle Gebiete eine interessante Auswertemöglichkeit dar, wo sich durch physikalische Parameter eine unzureichende Interpretierbarkeit von z. B. Störgeräuschen ergibt, diese jedoch deutlich mit dem Gehör wahrnehmbar sind.

Im Bereich der Mustererkennung stellt jeder der genannten Parameter ein Merkmal dar. Dabei kann jedoch auch noch unterschieden werden zwischen einwertigen (z.B. Effektivwert) und mehrwertigen Merkmalen (z.B. Amplitudenspektrum mit 512 Bändern, wobei jedes Band ein Merkmal bildet). Es ist jedoch klar, dass die Merkmale der Schlüssel einer erfolgreichen Erkennung bzw. Klassifizierung sind. Dabei müssen die Merkmale die z. B. zu trennenden Geräusche eindeutig beschreiben. Weiterhin sollten zur Reduktion des Dimensions- und Rechenaufwands nur eindeutig relevante Merkmale verwendet werden, die untereinander keine Redundanzen aufweisen.

## 2.5 Interpretation hochdimensionaler Daten

Die Identifikation der relevanten Merkmale kann empirisch oder automatisiert erfolgen. Die Interpretation der Merkmale/ Datenreihen wird auch als *Data Mining* bezeichnet. Zum einen werden durch Versuchsreihen, Vorwissen oder physikalisch technische Kenntnisse die Merkmale genutzt, welche das Geräusch sehr gut abbilden. Und zum anderen werden Algorithmen eingesetzt, die z.B. durch vorgegebene Signalanalyse-Prozesse die wichtigen Merkmale identifizieren. Gerade für sehr komplexe Vorgänge, wo eine Vielzahl von potentiellen Merkmalen vorliegen, eignet sich die automatisierte Variante. So werden beispielsweise in [Cha05, YLG04, YFH<sup>+</sup>08] Verfahren zur Reduktion und Bestimmung der Relevanz von Merkmalen vorgestellt und mit folgenden Begriffen umschrieben:

- *feature selection*
- *feature extraction*
- *feature ranking*
- *feature relevance*
- *feature weighting*
- *feature subset selection*
- *relevance learning*
- Merkmal-Relevanzbestimmung
- Merkmalreduktion

So werden in [BWJ06, wCJ07, SRD<sup>+</sup>04, MBGMF04] Klassifikatoren genutzt, um die schrittweise reduzierten Merkmalsätze anhand der Erkennungsrate zu bewerten. In [WM04, RvK03, YFH<sup>+</sup>08, FYH<sup>+</sup>08, DP05, PLD05, YLG04] werden hingegen Verfahren vorgestellt, die anhand von informationstheoretischen oder statistischen Größen die Merkmale hinsichtlich Relevanz und Redundanz bewerten. Es zeigt sich, dass dieses Gebiet in einem sehr großen Bereich Anwendungen findet. Oftmals bestehen Datensätze aus der Biomedizin - z.B. zur Analyse von Krebsfällen - aus mehreren hundert oder tausend Merkmalen (Genen). Im Bereich der Bioinformatik werden diese Datensätze analysiert, um Rückschlüsse auf Ursachen und Wirkung zu erlangen. Eine empirische Interpretation der Daten ist bei diesen hochdimensionalen Daten sehr aufwendig, weshalb eine automatisierte Bewertung bevorzugt wird. Weitere Anwendungsgebiete von automatisierten Methoden zur Interpretation hochdimensionaler Datensätze sind u. a. in der Finanzmathematik (Kreditwürdigkeit, *Credit Scoring* [HEB08, HBV05]) oder Musiktitererkennung (MPEG-7 Audio- und Videodescriptoren, [MSS02, Roh03, Sch04]) gegeben.

Sofern Merkmale identifiziert sind, die die zu trennenden Geräusche eindeutig beschreiben, wird durch ein Klassifikationsverfahren die Eigenschaft der Merkmale analysiert und ausgewertet. Dabei werden je nach Anwendungsgebiet verschiedenste Verfahren genutzt, welche sich grob in

- überwachte und
- unüberwachte

Algorithmen einteilen lassen. Diese unterscheiden sich in der Lernphase der Merkmaleigenschaften. Ein überwachtes Lernverfahren analysiert die Merkmale in Abhängigkeit vorgegebener Klasseninformationen. Ein unüberwachtes Verfahren hingegen interpretiert die Daten selbst und bildet deren Struktur ab, ohne weitere Informationen über die Klassenzugehörigkeit zu bekommen. In den in Abschnitt 2.3 genannten Veröffentlichungen wurden meist einfache überwachte Lernverfahren, wie z.B. Schwellenwertklassifikatoren in [Guo05, Pol05] zur Umsetzung eines akustischen Prüfsystems genutzt. Dennoch wurde beispielsweise in [FWH98] eine selbstorganisierende Karte verwendet, welche zwar eigenständig die Struktur der Merkmale adaptiert, die folgende Auswertung jedoch auch über vorgegebene Klasseninformationen erfolgt. Selbstorganisierende Algorithmen wurden schon ausgiebig erforscht, interessante Veröffentlichungen mit allgemeinen und anwendungsbezogenen Informationen sind [SG04, KK96, Kan94, LKP06]. Für die neuronale Karte existieren

verschiedene Varianten, siehe [KKL90]. Diese unterscheiden sich im Rechenaufwand, Parallelverarbeitungsmöglichkeit, Dimensionalität oder Echtzeitfähigkeit ([SM01, Sei02, Sei01]). Eine weitere interessante Anwendung dieser Algorithmen stellt die *Databionic MusicMiner* Software dar ([MUT<sup>+</sup>05]). Diese fungiert als Musikdatenbank-Browser und kategorisiert die auf der Festplatte befindlichen Audio-Dateien. Dafür werden Merkmale aus den akustischen Daten berechnet, welche unabhängig vom Künstlernamen oder Liedtitel sind. Der selbstorganisierende Algorithmus interpretiert diese hochdimensionalen Daten und erzeugt *MusicMaps*; diese visualisieren Ähnlichkeiten der Musikstücke/ Künstler.

## 2.6 Ausrichtung der Arbeit

Die aufgenommenen Signale (von Mikrofonen oder Beschleunigungsaufnehmern) werden transformiert, um Merkmale zu generieren, welche die Eigenschaften des Sensorsignals beschreiben. Dabei werden zum Teil psychoakustische Verfahren angewendet, welche für alle Gebiete eine interessante Auswertemöglichkeit darstellen, wo sich durch physikalische Parameter (wie Effektivwert, Fourier-Transformierte) eine unzureichende Interpretierbarkeit von z. B. Störgeräuschen ergibt, diese jedoch deutlich mit dem Gehör wahrnehmbar sind. Die Merkmale können empirisch oder automatisiert in ihrer Statistik analysiert werden, um deren Eignung für die Klassifikations-/ Mustererkennungsaufgabe zu untersuchen. Eine einfache Möglichkeit zur Bewertung der Trennbarkeit zweier Merkmale stellt beispielsweise die *Fisher*-Diskriminante (FD) dar. Die eigentliche Fehlererkennung erfolgt anhand der fehlerspezifischen Merkmale z. B. durch Schwellenwertklassifikatoren. Eine gute Interpretierbarkeit - beispielsweise gegenüber Neuronalen Netzen - weisen regelbasierte Klassifikatoren auf ([Guo05]). Eine interessante Methodik stellt auch die in [Wir00] vorgestellte kaskadierte Fehlererkennung dar, wo in der ersten Verarbeitungsstufe das Geräusch in "in Ordnung"/ "fehlerhaft" und in der nachfolgenden Stufe bei Bedarf der Fehler im Speziellen für "fehlerhafte" Geräusche klassifiziert wurde. Im Weiteren geht aus den Veröffentlichungen hervor, dass eine ausreichende Menge an Daten vorhanden sein muss, um eine statistische Sicherheit zu erreichen. In den meisten Fällen ist ein Datenbankaufbau notwendig. Auch die Klassifikation über eine selbstorganisierende Karte in [FWH98] ist hervorzuheben, da diese selbstständig die hochdimensionale Anzahl von Merkmalen interpretieren kann.

In allen betrachteten Diagnosekonzepten gehen folgende Punkte hervor, die in dieser Arbeit besonders betrachtet werden sollen:

- Wie reagiert ein akustisches Prüfsystem bei Verwendung eines Sensorsignals mit undefinierter akustischer Umgebung. In den meisten Fällen arbeiten die Diagnosesysteme in sehr definierten Umgebungen (z.B. Prüfstand in akustisch optimierter Kabine). Welchen Einfluss haben sich ändernde Umgebungsgeräusche/ -charakteristika auf die Merkmale?
- In wie weit kann eine akustische Datenbank zur gezielten Ablage von Geräuschbeanstandungen im Bereich der Qualitätssicherung für ein akustisches Diagnosesystem sinnvolle Ergebnisse bringen? Wie ist eine Datenbank sinnvollerweise zu strukturieren?

Weiterhin soll das Geräuschlexikon als Diagnosehilfe bei Volkswagen untersucht und optimiert werden. Dieses fördert bisher lediglich eine einheitliche Kommunikation über die Geräuscharten,

unterstützt jedoch bisher nicht maßgeblich die Werkstätten beim schnelleren Beheben von Fehlern. In dieser Arbeit sollen ebenfalls Möglichkeiten zur Einführung und Verbesserung einer akustischen Diagnose im Werkstattbereich erarbeitet werden.

## 3 Aufgabenstellung und Grundlagen der akustischen Mustererkennung

In diesem Kapitel soll das Konzept für die Entwicklung eines akustischen Diagnosesystems für den Einsatz im Fahrzeuginnenraum vorgestellt werden. Das Konzept wurde im Rahmen eines Projektes der Volkswagen AG Konzernforschung erarbeitet. Prinzipiell lässt es sich dadurch in einen praktischen (Hardwareaufbau, Abstimmung innerhalb des Konzerns usw.) und einen wissenschaftlichen Teil (Algorithmen- und Modellentwicklung) trennen. Der hier vorgestellte "Lösungsweg" soll vor allem die Untersuchung kritischer Randbedingungen ermöglichen und wenn möglich einige Realisierungsmöglichkeiten bieten. Einige Aspekte des vorgestellten Konzeptes sind an die im Patent [Opf09] genannten Ideen und Ansätze angelehnt.

Des Weiteren werden in diesem Kapitel auch erste Grundlagen zum Verständnis der folgenden Kapitel erörtert.

### 3.1 Konzepterstellung

Wie in Kapitel 2.3 beschrieben ist, werden die meisten akustischen Prüfsysteme bei definierten Randbedingungen eingesetzt. Da laut Vorgabe in dieser Arbeit ein Standard Messmikrofon im Fahrzeug-Innenraum zur Analyse verwendet werden soll, besteht ein Themengebiet des Konzepts (Abbildung 3.1) - zur Realisierung eines akustischen Prüfsystems - im Aufbau einer *Datenbank (SQL)*:

- Zur Ablage von Geräuschen und deren Protokoll-Daten.
- Dabei werden diverse Randbedingungen verändert. Es sollen Messungen bei unterschiedlichen Fahrzeugen, Wetterbedingungen (z.B. trocken, feucht), Fahrbahnbelägen, Fahrzuständen und vor allem verschiedenen Fehlergeräuschen durchgeführt werden. Für die Definition der Fehlergeräusche sollen in Rücksprache mit der Qualitätssicherung typische Fehlergeräusche und deren Ursache systematisch erfasst und analysiert werden.
- Für die aufgenommenen Geräusche sollen physikalische, akustische als auch psychoakustische Kenngrößen/ Merkmale berechnet und in einer Datenbank gespeichert werden.
- Durch Signalanalyse werden die relevanten Merkmale identifiziert. Hierfür ist eine Ausarbeitung und Umsetzung verschiedener Verfahren notwendig, welche anschließend anhand der Datenbank-Geräusche getestet werden. Diese Teilaufgabe stellt einen sehr anspruchsvollen Bereich dar, da aus bisherigen Analysen hervorgeht, dass die Merkmale den Schlüssel

für eine erfolgreiche Klassifikation darstellen. Die relevanten Merkmale bilden einen akustischen Fingerabdruck, welcher die markanten Eigenschaften aller untersuchten Geräusche widerspiegelt.

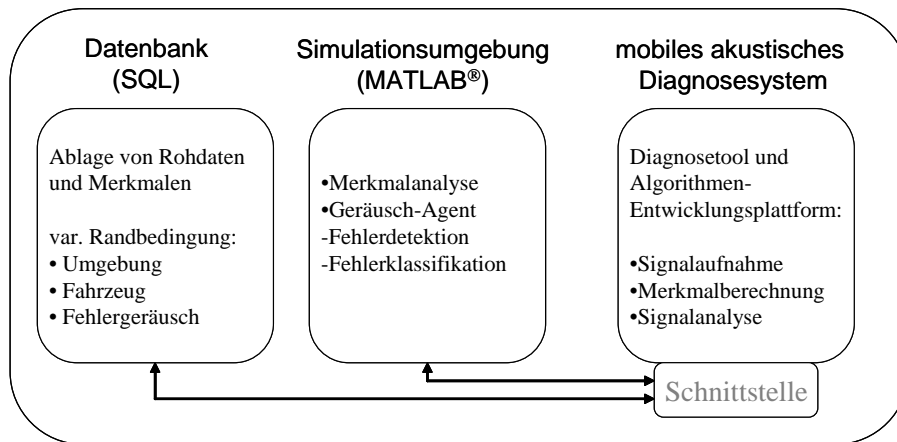


Abbildung 3.1: Konzept zum Aufbau eines akustischen Prüfsystems für den Fahrzeuginnenraum

In Abbildung 3.2 ist das Konzept zum Aufbau der Datenbank beschrieben, Fehlergeräusche (von beispielsweise Bauteilen im Vorderwagen) werden systematisch vermessen und abgespeichert. In [Wor08] wurden bereits erste Analysen zur Eignung von Messtechnik und zur strukturierten Ablage von Messdaten anhand eines Versuchsfahrzeugs durchgeführt.

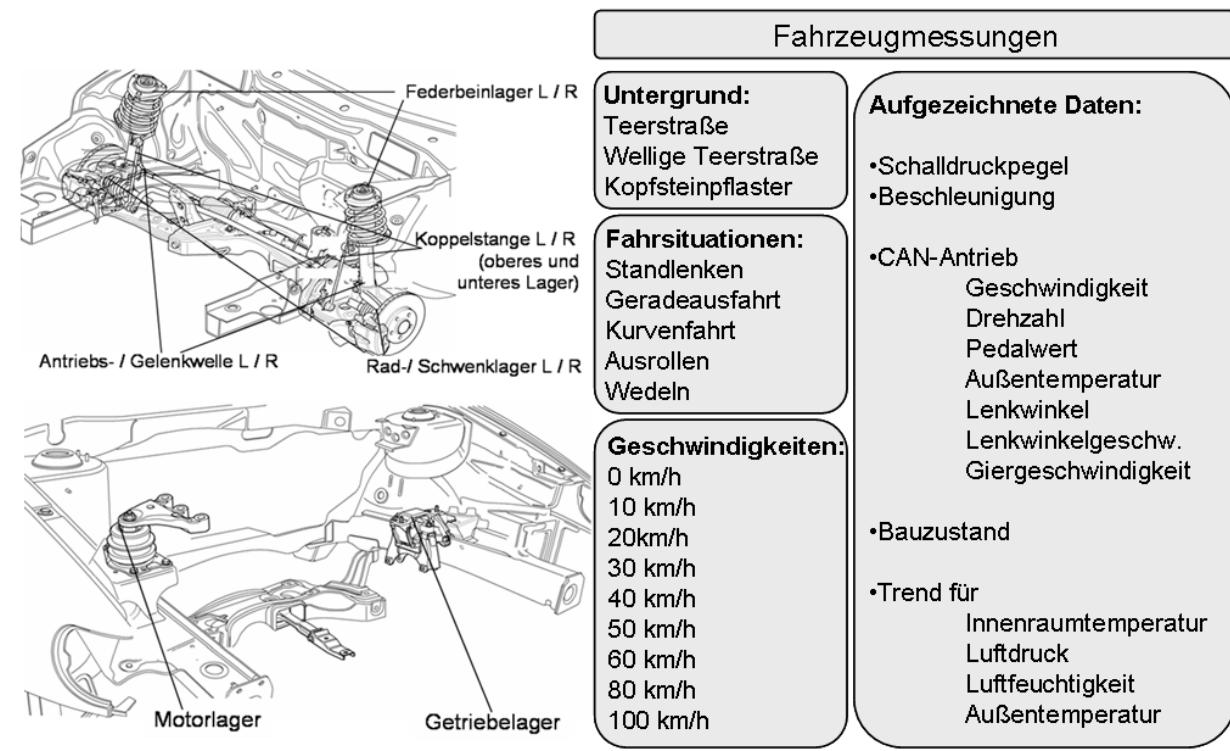


Abbildung 3.2: Beschreibung der Datenbankstruktur

Daraus ging hervor, dass ein Geräuschunterschied zwischen verschiedenen Fehlergeräuschen und dem fehlerfreiem Zustand wahrzunehmen ist. Weiterhin treten die Geräusche in typischen Frequenzbereichen auf. Dennoch konnten anhand 1292 Messungen nicht alle Einflussgrößen abgebildet werden und “entsprechend dieser Überlegungen sind diese Messergebnisse als Test und Vergleichsdaten für erste Analysealgorithmen nutzbar” (Zitat). Aufbauend auf diesen Erkenntnissen soll die Datenbank um weitere Messungen ergänzt werden. Dabei sollen zusätzlich zu den akustischen Sensorsignalen auch fahrzeugbeschreibende CAN-Daten aufgezeichnet werden.

Den nächsten Schwerpunkt im Konzept (basierend auf den akustischen Messdaten in der Datenbank) des Prüfsystems stellt die Merkmalanalyse und der *Geräusch-Agent* dar:

- Dieser soll angelehnt an [Wir00] in kaskadierter Form aufgebaut werden:
  - Modell zur Fehlerdetektion; erkennt, ob ein Geräusch im Fahrzeuginnenraum auf ein defektes Bauteil zurückzuführen ist. Erst wenn dies eintritt, wird die nächste Stufe ausgeführt.
  - Modell zur Fehlerklassifikation; unterscheidet, anhand des akustischen Fingerabdrucks, welches Bauteil dieses Geräusch verursacht hat.

Weiterhin sollen neben den Erkenntnissen der Merkmalanalyse auch selbstorganisierende Algorithmen, wie die selbstorganisierenden Karten nach [Koh01], in die Umsetzung des Agenten einfließen.

Wie aus diesem beiden Schwerpunkten hervorgeht, sind eine Vielzahl von Herausforderungen an deren Umsetzung gebunden. Hinsichtlich des Datenbankaufbaus müssen beispielsweise der enorme Speicherplatzbedarf von akustischen Messdaten betrachtet werden. Eine 24-Bit Audio-Datei bei einer Abtastfrequenz von 48 kHz benötigt für 60 Sekunden Aufnahmezeit ca. 8,2 MB.

In [Kot08] wurden anhand der ersten Daten von [Wor08] verschiedene Merkmale aus der Akustik, Physik, Psychoakustik berechnet und hinsichtlich ihrer Klassifizierbarkeit untersucht. Dabei wurde gezeigt, dass sich die Daten klassifizieren lassen, der Aufwand zur Bereitstellung der Merkmale und deren Reduktion jedoch sehr zeitaufwendig ist. Diese wurden bisher in verschiedenen Programmen berechnet und anschließend exportiert bzw. in der Simulationsumgebung importiert um beispielsweise die Reduktion durchzuführen. Aus diesem Grund sollen im Rahmen dieser Arbeit auf Basis der erweiterten Datenbank:

- effiziente Merkmalreduktionsverfahren getestet und dadurch möglichst wenige Merkmale zur Klassifikation genutzt werden,
- ein einfaches Programm zur Aufnahme, Berechnung, Export der Merkmale und Datenbankablage umgesetzt werden.

Bindeglied zwischen der Datenbank und der Simulationsumgebung, auf welcher die Algorithmen und Modelle entwickelt werden, stellt das mobile akustische Diagnosesystem in Abbildung 3.1 dar. Dieses soll neben der Geräuschaufnahme auch mit Schnittstellen zur Datenbank (z.B. *Structured Query Language*, SQL) und Matlab® umgesetzt werden, so dass neu entwickelte Algorithmen direkt im Fahrzeug getestet werden können. Basierend auf den Erkenntnissen und Analysen

beim Datenbankaufbau und Simulieren, sollen Anforderungen definiert und ein Diagnosesystem entwickelt werden.

Durch dieses klar definierte aber dennoch universelle Konzept wird nicht nur im Hinblick auf die zur Bearbeitung dieser Dissertation notwendigen Analysen eine Grundlage geschaffen, sondern auch ein Fundament für folgende Arbeiten im Bereich der akustischen Diagnose der Volkswagen Konzernforschung gelegt.

## 3.2 Mustererkennungsmodell

Viele Klassifikationsaufgaben lassen sich beispielsweise in die in Abbildung 3.3 gezeigten Teilaufgaben, unabhängig von genaueren Spezifikationen der vorliegenden Signale, Daten oder Vorwissen zerlegen. Bei den beschreibenden Größen wird zwischen Merkmalen, Mustern und Klassen unterschieden:

- **Merkmale:** Sind die beschreibenden Kenngrößen eines Vorgangs, beispielsweise die Farbinformation eines Pixels oder der Effektivwert eines Mikrofonsignals.
- **Muster:** Diese beschreiben die Ausprägung der Merkmale. Wenn beispielsweise mehrere grüne Äpfel digital fotografiert und die Merkmale (Pixel-Farbinformationen) analysiert werden, dann ergeben die Merkmale eines Apfels ein Muster, welches jedoch nicht mit einem anderen Apfel - aufgrund unterschiedlicher Farbgebung oder Größe - identisch ist.
- **Klassen:** Diese beschreiben die Zusammengehörigkeit der Muster. So könnten beispielsweise die Klassen "roter/ grüner" Apfel die Muster der Äpfel mit entsprechender Farbe enthalten.

Im ersten Schritt müssen die Daten gesammelt und in ein verarbeitungsfähiges Format gebracht werden. Bei großen Datenmengen müssen gegebenenfalls Datenbanken aufgebaut werden. Wichtig ist weiterhin, dass auf eine Vergleichbarkeit der Daten geachtet und beispielsweise gleiche Einheiten oder Wertebereiche bei der Speicherung der Daten verwendet werden.

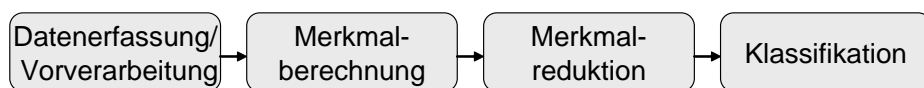


Abbildung 3.3: Vereinfachter Ablauf einer Klassifikationsaufgabe

Im nächsten Schritt werden die Merkmale für alle erfassten Daten berechnet. Für ein akustisches Signal könnten der Effektivwert oder die Frequenzwerte eines Amplitudenspektrums berechnet werden. Die Merkmale sollen die Daten möglichst detailliert beschreiben.

Die Merkmalreduktion bewertet anschließend die berechneten Merkmale und definiert die wichtigen Merkmale, welche einen wesentlichen Beitrag zur Identifikation einer vorgegebenen Klasse darstellen. Dadurch können der Rechenaufwand für nachfolgende Klassifikationsalgorithmen reduziert werden. Weiterhin können Merkmalreduktionsverfahren auch zur Analyse der Merkmale genutzt werden, wenn Unklarheit besteht, ob in den Merkmalen klassenspezifische Muster vorhanden sind.



Bei der Klassifikation wird prinzipiell zwischen überwachten<sup>1</sup> und unüberwachten<sup>2</sup> Verfahren unterschieden. Dabei variiert die Komplexität der Algorithmen und deren Interpretierbarkeit für beide Verfahren. Die Aufgabe des Klassifikators besteht darin, in einer Lern- bzw. Trainingsphase die gegebenen Muster zu interpretieren und seine eigene Struktur (z.B. anhand von Gewichten) danach auszurichten. Nach abgeschlossener Trainingsphase soll der Klassifikator neue bzw. unbekannte Muster bewerten und dadurch einen Zusammenhang zu den gelernten Ausprägungen der antrainierten Muster/ Klassen bilden.

### Konfusionsmatrix/ Wahrheitsmatrix zur Bewertung eines Klassifikators

Eine Konfusionsmatrix (oder Wahrheitsmatrix) dient zur Beurteilung eines Klassifikators. Dafür werden die tatsächlichen mit den durch den Klassifikator ermittelten Häufigkeiten des Auftretens der Klassen verglichen. In einer Tabelle (siehe Tabelle 3.1) werden - für alle möglichen Kombinationen von ermittelter Klasse und tatsächlicher Klasse - die Häufigkeiten eingetragen.

ERMITTELTE\TATSÄCHLICHE KLASSE	BLAU	ROT
BLAU	<i>rb</i> - richtig blau	<i>fb</i> - falsch blau
ROT	<i>fr</i> - falsch rot	<i>rr</i> - richtig rot

Tabelle 3.1: Konfusionsmatrix für einen Zweiklassenfall

Aus der Konfusionsmatrix kann direkt die Klassifikationsrate oder Erkennungsrate (im engl. *accuracy*) berechnet werden, welche zur Bewertung der Eignung eines Klassifikators verwendet wird:

$$acc = \frac{rb + rr}{(rb + fb + fr + rr)} \quad (3.1)$$

## 3.3 Überblick zum Aufbau der Arbeit

Gegenstand dieser Arbeit ist der Entwurf eines Systems zur Analyse des Fahrzeuginnenraumgeräuschs und der Detektion von unerwünschten Geräuschen, die durch Defekte oder verschlissene Bauteile verursacht werden. Das Interesse der Volkswagen AG Konzernforschung liegt in der Minderung von Kosten - die beispielsweise durch Bauteilfehltausch entstehen - und der Verbesserung des Serviceprozesses (und den damit geringeren Wartezeiten für den Kunden) in den Werkstätten. In Zusammenarbeit mit der Qualitätssicherung werden kontinuierlich Geräusche erfasst bzw. analysiert (beispielsweise in Dauerlauffahrzeugen, Kundenfahrzeugen oder Entwicklungsträger mit Prototypenbauteilen) und daraus Verbesserungen abgeleitet. Deshalb sind viele wissenschaftliche Arbeiten in der Konzernforschung sehr praxisbezogen, was auch für diese Arbeit im Rahmen der Industriepromotion gilt.

<sup>1</sup>Zu jedem Muster erhält der Klassifikator im Lernvorgang eine Klasseninformation.

<sup>2</sup>Der Klassifikator lernt anhand der gegebenen Muster ohne Klasseninformation.

In den Kapiteln 1, 2 und 3 wurde das Thema der automatischen Diagnose eingeführt, Begriffe erläutert, bekannte Systeme und Realisierungsansätze analysiert und ein Konzept erstellt. Daraus ergeben sich mehrere Punkte, die besonderes Interesse bei der Bearbeitung der Aufgabe wecken:

- Idee des fahrzeugspezifischen akustischen Fingerabdrucks:  
Es sollen Merkmale in Form eines akustischen Fingerabdrucks definiert werden, welche die Mustererkennung im beschriebenen Umfeld ermöglichen.
- Umsetzung einer akustischen Gesamtsystemdiagnose:  
Dabei soll dieses nicht einzelne Bauteilgeräusche, sondern einen Verbund aus mehreren stark variierenden Fehlergeräuschen von diversen Bauteilen/ Strukturen und Umgebungsgeräuschen diagnostizieren.
- Messdaten werden im Fahrzeuginnenraum erfasst:  
Dies stellt eine besondere Herausforderung dar, da keine künstlich optimierte/ definierte Umgebung vorliegt. Vor allem im Bereich der Merkmalreduktion wird sich zeigen, ob diese Aufgabe durch ein Mustererkennungsverfahren bewältigt werden kann.
- Einsatz eines unüberwachten Lernverfahrens:  
In den Recherchen wurde ermittelt, dass die Anwendung von selbstorganisierenden Karten für die Interpretation mehrdimensionaler Daten geeignet scheint. Bisher gibt es nur vereinzelte Veröffentlichungen, die die Anwendung der SOM bezogen auf akustische Daten behandeln.
- Lokalisation des Fehlergeräusches:  
Anhand der These, dass die Ausbreitung des Schalls für jedes Bauteil einzigartig ist und sich dies in dem akustischen Fingerabdruck widerspiegelt, soll auch eine einkanalige Analyse Aufschluss über den Ort der Fehlergeräusch verursachenden Schallquelle liefern.
- Einheitliche Fingerabdruckberechnung für alle Fahrzeugtypen:  
Es ist zu klären, ob ein allgemeingültiger Fingerabdruck für verschiedene Fahrzeugtypen berechnet werden kann, welcher beispielsweise eine Identifikation typischer Fehlergeräusche ermöglicht.

In Kapitel 4 werden zwei große Themengebiete erläutert.

Zum einen die Klassifikation. Dabei wird auf die Verarbeitung der akustischen Daten eingegangen und ein Überblick über häufig angewandte Verfahren erarbeitet. Wichtige Klassifikationsverfahren für folgende Kapitel werden ebenfalls näher erläutert, wie Neuronale Netze bzw. selbstorganisierende Karten. Ein Teilbereich dieses Themengebiets behandelt auch die Umsetzung eines transparenten und recheneffizienten Klassifikators und deren Analyse für Steuergeräteimplementationen.

Zum anderen wird die Merkmalselektion beschrieben. Dabei werden die Verfahren anhand ihrer Eigenschaften eingeordnet, da eine Vielzahl mathematischer Verfahren aus der Signalverarbeitung, Statistik und Informationstheorie zur Interpretation von Datenmengen verwendet werden können. Weiterhin wird detailliert auf ausgewählte Algorithmen - zur Identifikation von Redundanz und Relevanz im mehrdimensionalen Merkmalraum - eingegangen.

Im Kapitel 5 folgen die Analysen der diversen Merkmalreduktionsverfahren. Dazu wird vorerst auf die Strukturierung der Testdaten bzw. Merkmalsätze zur Generierung von Datensätzen und deren Vorverarbeitung eingegangen. Anschließend werden diverse Selektionsverfahren systematisch getestet und bewertet. Die besten Verfahren werden identifiziert und damit die akustischen Fingerabdrücke generiert. Ein weiterer wichtiger Abschnitt behandelt den Vergleich der Fingerabdrücke zwischen verschiedenen Fahrzeugtypen im Hinblick auf die Überprüfung der Allgemeingültigkeit.

Aufbauend auf diesen Erkenntnissen wird im Kapitel 6 die Entwicklung eines Agenten zur automatischen Geräuschanalyse im Fahrzeuginnenraum vorgestellt. Es werden zwei Evaluierungsschritte gezeigt, erstens die *Korrelationsmethode zur Trennung akustischer Zustände* und zweitens die *überwachte akustische Diagnose mit Self-Organizing-Maps* mit dem Ziel der Entwicklung eines funktionsfähigen Agenten. Es werden Parameteranalysen durchgeführt und dadurch die Algorithmen optimiert. Der Test und Funktionsnachweis erfolgt anhand einer optimierten Stichprobe, aber auch auf die gesamte Datenbank mit insgesamt mehreren hundert Aufnahmen pro Fahrzeug. Dabei ist hervorzuheben, dass im Durchschnitt jede Aufnahme 25 bis 30 Sekunden lang ist und schlussfolgernd die Validierung an ca. 500 Gigabyte Akustikdaten stattfindet.

Die Entwicklung eines mobilen akustischen Diagnosesystems wird in Kapitel 7 beschrieben. Dabei sollen die Erkenntnisse und Algorithmen vorangegangener Kapitel in einem mobilen System zur akustischen Signalerfassung vereint werden. Anforderungen aus der Qualitätssicherung werden ebenfalls berücksichtigt. Es wurden zwei Entwicklungsschritte definiert, die in diesem Kapitel vorgestellt werden.



## 4 Merkmalreduktion und Klassifikation

Aus dem Konzept, welches im Kapitel 3 vorgestellt wird, geht hervor, dass sowohl die Merkmalsberechnung und -reduktion als auch die Klassifikation wesentliche Punkte der Umsetzung eines Systems zur akustischen Geräuschanalyse darstellen. Aus diesem Grund wird in diesem Kapitel ein Abriss über verwendbare Algorithmen und Verfahren zur Bearbeitung der genannten Schwerpunkte gegeben, mit dem Ziel, einen akustischen Fingerabdruck zu bilden, welcher als Grundlage für einen Klassifikator dient. Dieser ermöglicht die Interpretation des aufgenommenen Fahrzeuginnenraum-Geräusches und die Zuordnung zu einer Geräuschkategorie, welche das Fehlerbauteil oder den Originalzustand definiert.

### 4.1 Physikalische Kenngrößen als Merkmale

Bekannte physikalische Parameter, die aus den aufgenommenen Mikrofonsignalen im Fahrzeuginnenraum berechnet werden, sind:

- **Fast-Fourier-Transformation (FFT)**

Ermöglicht die Berechnung des Frequenzspektrums. Stellt eine schnelle Berechnungsvariante der Diskreten Fourier-Transformation (DFT) dar, siehe [Mey06].

- **Spectral Flux** (Spektrale Schwankungsstärke)

Gibt die Veränderung eines Spektrums über den Zeitverlauf an.

$$SF(t) = \sum_{n=1}^N (M_N(n, t) - M_N(n, t-1))^2 \quad (4.1)$$

- **Spectral Centroid** (Spektraler Schwerpunkt)

Bewertet den Schwerpunkt eines Spektrums.

$$SC = \frac{\sum_{n=1}^N M(n) f(n)}{\sum_{n=1}^N M(n)} \text{ [Hz]} \quad (4.2)$$

- **Spectral Flatness**

Bewertet die Verteilung der Energie eines Spektrums. Umso mehr die Werte der Frequenzbänder Gleichverteilung aufweisen, desto höher wird die *SFL*.

$$SFI = \frac{\sqrt[N]{\prod_{n=1}^N M(n)}}{\frac{\sum_{n=1}^N M(n)}{N}} \quad (4.3)$$

- **Spectral Rolloff**

Berechnet wird die Rolloff-Frequenz, bis zu der sich  $\alpha\%$  der Frequenz-Verteilung aufsummieren.

$$\sum_{n=1}^{SRf} M(n) = \frac{\alpha_{SRf}}{100} \sum_{n=1}^N M(n) \quad (4.4)$$

- **Cepstrum**

Berechnet die inverse Fourier-Transformation aus dem logarithmierten Betragsspektrum.

$$c_x = FT^{-1} [\log (|FT(x(t))|)] \quad (4.5)$$

- **Mel-Frequenz-Cepstrum-Koeffizienten (MFCC)**

Besteht aus zwei Berechnungsschritten und wird auch Mel-Cepstrum genannt:

- Mel-Skala Filterbankverarbeitung zur Zerlegung des Signals in Frequenzbänder.
- Diskrete Kosinus-Transformation (DCT) der Frequenzbänder.

- **Scheitelfaktor (Crest-Factor, C)**

Beschreibt den Verlauf eines Signals. Der Rückgabewert steigt mit zunehmenden Verzerrungen/ Spitzen im Signal  $x$ .

$$C = \frac{x_{max}}{x_{eff}} \quad (4.6)$$

- **Effektivwert (Root-Mean-Square, rms oder RMS)**

$$rms = \sqrt{\frac{1}{T} \sum_{t=1}^T x(t)^2} \quad (4.7)$$

- **Oktavspektrum**

Einteilung des Spektrums in Oktavbänder (siehe Anhang A.1).

- **Terzspektrum**

Einteilung des Spektrums in Terzbänder (siehe Anhang A.1).

- **Diskrete Kosinus-Transformation**

Ähnlich der DFT handelt es sich um eine diskrete Frequenztransformation, siehe [Wer08]. Bei der DCT ist hervorzuheben, dass diese einen Großteil der Signalenergie und damit der

wesentlichen Informationen in nur einigen wenigen Koeffizienten abbilden kann. Aus diesem Grund wird dieses Verfahren beispielsweise zur Datenreduktion bei der Bildverarbeitung eingesetzt.

- **Diskrete Wavelet-Transformation (DWT)**

Ist eine Wavelet-Transformation, jedoch zeit- und frequenzdiskret. Als Berechnungsfunktionen werden die Wavelets verwendet. In [Stö09] ist eine kleine Übersicht zur Berechnung gegeben. Theorie und Anwendungen werden ausführlich in [LMR98] behandelt.

- **Nulldurchgang-Rate (Zero-Crossing Rate, ZCR)**

Diese Berechnungsvorschrift ermittelt die Anzahl der Vorzeichenwechsel für einen definierten Zeitbereich  $T$  des Signals  $x$ .

$$ZCR = \frac{1}{T} \sum_{t=2}^T \prod \{x(t) x(t-1) < 0\} \quad (4.8)$$

- **Kurtosis (Schiefe, Wölbung)**

Die Kurtosis bewertet die Verteilung einer Zufallsgröße  $x$ . Eine spitz zulaufende Verteilung erzeugt eine hohe Kurtosis, eine flache Verteilung hingegen einen niedrigen Rückgabewert.

$$Kurtosis = \frac{\frac{1}{T} \sum_{t=1}^T (x(t) - \bar{x})^4}{\left(\frac{1}{T} \sum_{t=1}^T (x(t) - \bar{x})^2\right)^2} \quad (4.9)$$

- **Modulationsfrequenz (Modulation Frequency)**

Bezeichnet die Frequenz der Amplitudenänderung bei einer Amplitudenmodulation [HEA05].

- **Artikulationsindex/ Sprachverständlichkeit (Articulation Index, AI)**

Wird genutzt, um die Verständlichkeit von Sprache zu bewerten. Der Rückgabewert befindet sich im Bereich  $0 \leq AI \leq 1$  oder  $0\% \leq AI \leq 100\%$ . Je größer der  $AI$ , desto besser kann eine Person die Sprache hören/ wahrnehmen. Der  $AI$  wird beispielsweise bei der Beurteilung von Sprachkommunikationssystemen im Automobil eingesetzt [OBS07]. Eine Umsetzung des Algorithmus' erfolgt anhand den Veröffentlichungen von [Ber47].

## 4.2 Psychoakustische Kenngrößen als Merkmale

Die Psychoakustik beschreibt mit ihren gehörgerechten Parametern den Zusammenhang zwischen der Empfindung des Schalls und den physikalischen Schallfeldgrößen wie Pegel, Frequenz oder Modulationsgrad. Sie stellt also für alle Gebiete eine interessante Auswertemöglichkeit dar, wo sich durch physikalische Parameter eine unzureichende Interpretierbarkeit von z. B. Störgeräuschen ergibt, diese jedoch deutlich mit dem Gehör wahrnehmbar sind ([FZ07]). Es gibt bereits Analysensysteme, beispielsweise von Fa. *HEAD acoustics* [HEA10] oder Fa. Brüel & Kjaer [Brü10a],

welche psychoakustische Analysen softwareseitig ermöglichen. Bei den Recherchen hat sich gezeigt, dass nicht alle psychoakustische Berechnungsverfahren genormt sind. So gibt es für die Lautheit eine DIN [Lou91] Berechnungsvorschrift, aus der auch die Schärfe berechnet werden kann. Parallel existieren jedoch auch Berechnungsvorschriften nach [Aur85a]. Für die Rauigkeit und Schwankungsstärke existieren bisher keine genormten Berechnungsvorschriften. In diesem Abschnitt werden die gängigsten psychoakustischen Parameter eingeführt, weitere detaillierte Informationen zur Theorie und Interpretation sind in [Mül04, Mös10] oder zusammengefasst in [Brä10b] gegeben.

- **Schwankungsstärke und spezifische Schwankungsstärke** (*Fluctuation Strength*)

Die Schwankungsstärke bestimmt die zeitliche Änderung der Lautstärke mit Modulationsfrequenzen bis ca. 20 Hz. Als Einheit der Schwankungsstärke wurde das *vacil* definiert, welches der Schwankung eines 1-kHz-Tones bei einem Schalldruck von 60 dB, mit einer Modulationsfrequenz von 4 Hz und einem Modulationsgrad von 100 % entspricht. Erfolgt die Berechnung nicht über das gesamte Signal, sondern über einzelne Frequenzgruppen der Bark-Skala [HEA06], so spricht man von der in *vacil/bark* angegebenen spezifischen Schwankungsstärke.

- **Lautheit und spezifische Lautheit** (*Loudness*)

Beschreibt die subjektive Wahrnehmung der Lautstärke. Das Empfinden wird dabei linear auf einer Skala abgebildet mit der Einheit *sones*. Ein *sones* entspricht einem Pegel von 40dB bei einem Sinuston der Frequenz 1 kHz. Die Umsetzung der Lautheit erfolgt anhand den Vorgaben in [Lou91] für stationäre Signale. Für zeitvariante Signale wurde diese Norm in [Lou08] erweitert. Wird die Lautheit über die Frequenzgruppen der Bark-Skala [HEA06] berechnet, so entspricht dies der spezifischen Lautheit.

- **Rauigkeit und spezifische Rauigkeit** (*Roughness*)

Diese wird besonders in frequenz- und amplitudenmodulierten Schallen wahrgenommen, ab ca. 15 Hz bis 70 Hz Modulationsfrequenz. Bis 300 Hz fällt diese ab, und darüber ist keine Rauigkeit wahrnehmbar. Ein *asper* - die Maßeinheit der Rauigkeit - entspricht einem zu 100 Prozent mit 70 Hz modulierten Sinuston von 60 dB SPL<sup>1</sup>. Laut [Ter98] "sind das schnarrende beziehungsweise brummende Geräusch eines fliegenden Käfers und eines Elektro- oder Kolben-Motors, sowie amplituden- beziehungsweise frequenzmodulierte Töne mit Modulationsfrequenzen zwischen ungefähr 30 und 300 Hz" Beispiele für Schalle, welche Rauigkeit hervorrufen. Die spezifische Rauigkeit wird ebenfalls über die Frequenzgruppen der Bark-Skala berechnet. Für die berechneten Merkmale wird das Modell von [Aur85b] verwendet, wobei es auch ein Modell von [Sot94] gibt.

- **Schärfe** (*Sharpness*)

Die Schärfe ist angelehnt an die Wahrnehmung der spektralen Charakteristik eines Signals. Ein hochfrequentes Signal erzeugt eine hohe Schärfe. Diese wird durch die Einheit *acum* beschrieben. 1 *acum* entspricht einem schmalbandigen Rauschen der Mittenfrequenz von 1 kHz mit einer Bandbreite von weniger als 160 Hz und einem Pegel von 60 dB. Die Realisierung

---

<sup>1</sup>dB = Dezibel, SPL = *Sound Pressure Level*/ Schalldruckpegel



der Berechnung erfolgt anhand der Veröffentlichung von [Aur85a], wobei in den letzten Jahren auch eine Norm erarbeitet wurde, siehe [Sch09].

- **Tonhaltigkeit** (*Tonality*)

Die Tonhaltigkeit oder Tonalität bestimmt, wie stark ein Geräusch als tonal wahrgenommen wird. So werden Klänge (bestehen aus einem Grundton und Obertönen) als weniger tonal wahrgenommen als Einzeltöne, was sich auch in dem Rückgabewert der Tonalität widerspiegelt [Mös07]. Diese ist einheitenlos, und der Rückgabewert befindet sich im Bereich 0 bis 1. Ein Einzelton erzeugt frequenzunabhängig eine Tonalität von 1. Schmalbandiges Rauschen wird je nach Frequenzbandbreite bewertet, die Frequenzbreiten von 0.1, 0.2, 0.57 und 1.5 Bark entsprechen beispielsweise einer Tonalität von 0.6, 0.3, 0.2 und 0.1. Die Umsetzung des Algorithmus zur Berechnung der Tonalität erfolgt anhand [HEA07] und [TSS82].

### 4.3 Merkmalselektion: Erzeugung akustischer Fingerabdruck

Hauptaufgabe der Merkmalreduktion bzw. -selektion ist die Generierung eines akustischen Fingerabdrucks. Dieser besteht aus Merkmalen, welche das normale als auch ein auffälliges Innenraumgeräusch (z.B. durch defekte Bauteile verursacht) eindeutig beschreiben. Der akustische Fingerabdruck  $m_{out}$  muss dementsprechend aus Merkmalen (engl. *features*) bestehen, die eindeutig relevant für die Unterscheidung der Geräusche sind. Da sich prinzipiell beliebig viele Merkmale aus den diskretisierten Sensordaten (z.B. Mikrofone oder Beschleunigungsaufnehmer) berechnen lassen, deren Informationsgehalt bezogen auf die Mustererkennungsaufgabe jedoch stark variiert, werden die redundanten und irrelevanten Merkmale extrahiert (Abbildung 4.1). Dabei erfolgt die Berechnung eines neuen Merkmalvektors  $m_{out}$  aus dem Eingabevektor  $m_{in}$  durch ein Reduktionsverfahren mit dem Ziel  $d < d'$ .  $A^T$  stellt die anhand von Merkmalanalyse (Informationsgehalt, Relevanz, Redundanz) definierte Auswahlmatrix dar.



Abbildung 4.1: Verfahren zur Merkmalreduktion

Durch diese Dimensionsreduktion des Merkmalraums wird die Klassifikationsaufgabe hinsichtlich des Rechenaufwands minimiert und die Erkennungsrate der Klassen optimiert.

#### Problemstellung

Über die systematische Auswertung des Informationsgehalts der Merkmale kann eine Reduktion des Merkmalsatzes (engl. *subset*) erreicht werden [Hal99, GE03, LM98, GKWZ07, Wie02]. Darüber hinaus können durch die Auswertung der Häufigkeit von Merkmalen, deren Verteilung auf die Muster und deren Verteilung auf die Klassen weitere Informationen über die Relevanz der Merkmale gesammelt werden. Dabei wird beispielsweise analysiert, wie häufig ein Merkmal in verschiedenen Klassen auftritt. Ein Merkmal, welches gleichmäßig über alle Klassen verteilt ist, hat

einen geringen Informationsgehalt. Ein Merkmal, welches sich in den Mustern einer Klasse sehr stark zu den Mustern der anderen Klassen abhebt, wird hingegen ein großer Informationsgehalt zugesprochen.

Verschiedenste Algorithmen sind zur Selektion und Bewertung anwendbar [GKWZ07, LM98, YP97, GBNT04]. In [GE03] werden umfassend die Methoden der Merkmalreduktion beschrieben. Daraus geht auch hervor, dass die Algorithmen zur Merkmalsbewertung trotz der Vielfalt in drei Gruppen eingeteilt werden.

- **Filter-Methoden**

Ein Algorithmus durchsucht den Merkmalraum nach einem optimalen Merkmal-Satz. Als Entscheidungsfunktion werden einfache *Filterfunktionen* genutzt. Diese basieren auf Vorwissen, wie beispielsweise: "Die Merkmale müssen stark zu den Klassen korreliert sein.". Die Filterfunktion könnte dabei durch den Korrelationskoeffizienten oder der Transinformation (statistisches Maß zur Bewertung des Zusammenhangs zweier Variablen, siehe Abschnitt 4.3.6.2) implementiert werden. Bei den meisten Filter-Methoden wird demzufolge die Relevanz der Merkmale durch statistische Tests bewertet und anhand einer Rangliste der optimale Merkmalsatz erzeugt.

Vorteile:

- Algorithmen sind einfach zu implementieren
- Sind robust gegenüber Überanpassung (*overfitting*)
- Geringer Rechenaufwand

Nachteile:

- Stehen in keinem Verhältnis zu den Klassifikatoren und könnten demzufolge nicht brauchbar sein.
- Es ist kein *Feedback*-Zweig vom Klassifikator vorhanden.

Verfahren/ Algorithmen:

- Korrelationskoeffizient [TOB09]
- *Fisher*-Diskriminante [Kot08]
- *Mutual Information* [PLD05]
- *Relief*-Algorithmus [GBNT04]
- *FOCUS*-Algorithmus [AD91]

- **Wrapper-Methoden**

Diese sind ähnlich zu den Filter-Methoden. Der Grundgedanke besteht darin, jeden möglichen reduzierten Merkmalsatz zu testen. Der Test der Merkmale erfolgt jedoch nicht auf einfachen Filterfunktionen, sondern anhand einer schrittweisen Evaluierung an einer Klassifikationsaufgabe, siehe Abbildung 4.2 und [KJ97].

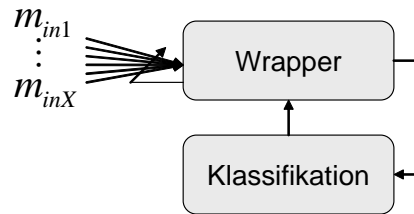


Abbildung 4.2: Alle möglichen Merkmalsätze  $m_{in1}$  bis  $m_{inX}$  werden durch den Wrapper in Kombination mit dem Klassifikator getestet.

Die Wrapper-Methoden sollen demnach die Brauchbarkeit der Merkmalsätze bewerten.

Vorteile:

- Optimaler und damit am besten geeigneter Merkmalsatz für das beste Klassifikationsergebnis wird ermittelt

Nachteile:

- Merkmalsatzgenerierung muss parallel zur Klassifikation erfolgen
- Rechenintensiv
- Gefahr der Überanpassung

Verfahren/ Algorithmen:

- Gütefunktion: Basiert meist auf Validierung, Kreuzvalidierung oder *leave-one out*.
- Die Suche erfolgt beispielsweise durch *forward selection*, *backward elimination*, *hill-climbing* oder genetische Algorithmen, siehe [KS95, SSnH98, RKFM02].

• **Embedded-Methoden**

Diese sind in ein Mustererkennungsmodell eingebettet und für dieses optimiert. Deshalb sind diese problemspezifisch ausgelegt und nicht übertragbar. Dabei orientiert sich der Aufbau an der Wrapper-Methode, jedoch ist der Algorithmus zur Merkmalsbewertung und -reduktion mit in den Lernvorgang des Klassifikators eingebettet [LCWE06].

Vorteile:

- Merkmalsatz wird in Abhängigkeit des Klassifikationsalgorithmus' ermittelt.
- Optimaler und damit bester Merkmalsatz für das genaueste Klassifikationsergebnis wird ermittelt.
- Weniger Rechenaufwand und geringere Gefahr der Überanpassung als bei Wrapper-Methoden.

Nachteile:

- Modularität nicht gegeben, Merkmalauswahl nicht bei anderen Klassifikatoren anwendbar.

### Verfahren/ Algorithmen:

- Cart (*Classification and Regression trees*) [BFOS84]
- ID3 [Qui86]
- C4.5 [Qui93]
- LASSO (*Least Absolute Shrinkage and Selection Operator*) [Tib94]

### 4.3.1 PCA-Hauptkomponentenanalyse

Die PCA (*Principal Component Analysis*) nach [Jol86] ist ein Verfahren aus der Statistik, welches zur Strukturierung und Vereinfachung hochdimensionaler Merkmalsätze eingesetzt wird. Dabei sollen die Vielzahl der Merkmale durch möglichst bedeutungsvolle Hauptkomponenten abgebildet werden. In [Smi02] wird das Verfahren sehr detailliert vorgestellt und verschiedene Beispiele erläutert. Dafür werden zuerst die mathematischen Grundlagen der Standardabweichung, Varianz, Eigenvektoren und Eigenwerte vorgestellt. Anschließend erfolgt die Berechnung der PCA (zur Identifikation von Mustern und deren optimierte Darstellung im transformierten Merkmalraum) zum Hervorheben von Unterschieden und Gemeinsamkeiten, in folgenden Schritten:

- Schritt 1: Datensatz erzeugen.
- Schritt 2: Mittelwert beseitigen.
- Schritt 3: Kovarianz-Matrix berechnen.
- Schritt 4: Eigenvektoren und Eigenwerte der Kovarianzmatrix berechnen.
- Schritt 5: Auswahl der Komponenten und Bilden eines Komponentenvektors.
- Schritt 6: Transformierten Datensatz bilden.

Die PCA wird in der Dimensionsreduktion und Clusteranalyse<sup>2</sup> angewendet. Neben der gerade vorgestellten linearen Realisierung der PCA existieren auch nichtlineare Hauptkomponenten Analyseverfahren die beispielsweise auf neuronalen Netzen basieren [Sch06].

### 4.3.2 Fisher Diskriminante

*Fisher's linear discriminant* nach [Fis36] dient zur Beschreibung der Separierbarkeit von zwei Stichproben von Objekten. In [Kot08] geht hervor: "Voraussetzung für die Nutzung dieser Kennzahl ist, dass sich die Verteilungen des Merkmals in den benutzten Stichproben als Gauß'sche Verteilungsfunktionen interpretieren lassen, also anhand ihrer Mittelwerte und Streuungen beschreibbar sind. Zur Berechnung der *Fisher-Diskriminanten*..." *FD* "...werden die Mittelwerte und Streuungen der zwei Objektklassen folgendermaßen ausgewertet". In [Ste99] ist folgende Berechnungsvorschrift gegeben:

---

<sup>2</sup>Die Clusteranalyse beschreibt strukturentdeckende, mehrdimensionale Analyseverfahren zur Ermittlung von Gruppen (*Clustern*) von Objekten mit gleichen oder ähnlichen Eigenschaften oder Ausprägungen.

$$FD = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \quad (4.10)$$

mit ,

$\mu_1, \mu_2$  - Mittelwert des betrachteten Merkmals für die Klassen 1, 2 ,

$\sigma_1, \sigma_2$  - Streuung des betrachteten Merkmals in den Klassen 1, 2.

Weitere Untersuchungen hinsichtlich Interpretation des  $FD$ -Wertes zur Auswirkung von Mittelwert und Streuung wurden in [Kot08] durchgeführt. Prinzipiell gilt, umso höher der  $FD$ -Wert, desto besser lassen sich die zwei Klassen voneinander trennen. Es fällt auf, dass die *Fisher*-Diskriminante die lineare Trennbarkeit von nur zwei Klassen untersucht und die Anwendbarkeit auf mehrere Klassen in Gleichung 4.10 nicht gegeben ist. Weiterhin ist ein Schwachpunkt, dass bedingte Informationen/ Zusammenhänge - die sich beispielsweise aus bestimmten Kombinationen von Merkmalen in Bezug auf ein anderes Merkmal ergeben - nicht analysiert werden.

### 4.3.3 F-Ratio

Eine erweiterte Form der Gleichung 4.10, bezogen auf mehrere Klassen, stellt die *F-Ratio* dar. In [Ste99] ist diese für  $n$  Stichproben und  $k$  Klassen wie folgt definiert:

$$FR = \frac{\frac{1}{k-1} \sum_{j=1}^k (\mu_j - \bar{\mu})^2}{\frac{1}{k} \frac{1}{n-1} \sum_{j=1}^k \sum_{i=1}^n (x_{ij} - \mu_j)^2} \quad (4.11)$$

mit

$\bar{\mu}$  - Mittelwert über gesamten Datensatz,

$\mu_j$  - Mittelwert aller Muster in Klasse  $j$ ,

$x_{ij}$  -  $i$ -te Messung in Klasse  $j$ .

Zusammengefasst stellt  $FR$  den Quotienten aus der Streuung der Mittelwerte der Klassen zu den Mittelwerten der Streuungen in den Klassen dar. Je höher der  $FR$ -Wert, desto besser ist ein Merkmal im Mittel über alle Klassen für die Trennung geeignet. Es ist zu beachten, dass ein Merkmal mit einer hohen Ausprägung aus diesem Grund nicht zwingend für alle Klassen die beste Trennbarkeit erreicht. Auch bei diesem Verfahren wird die Trennungswirkung von kombinierten Merkmalen nicht berücksichtigt.

### 4.3.4 Korrelationskoeffizient

Die Varianz einer Zufallsvariable  $\delta^2(x) = \delta_{xx}$  stellt ein Maß für die Streuung von ihrem Erwartungswert  $E(x)$  dar. Bezogen auf eine weitere Zufallsvariable  $y$  kann der nichtstandardisierte Zusammenhang mit  $x$  durch die Kovarianz  $\delta_{xy}$  berechnet werden. Für eine Auswertbarkeit ist eine gleiche Maßzahl für  $x$  und  $y$  Voraussetzung. Der Korrelationskoeffizient  $r$  [RN88] mit dem Wertebereich  $r = [-1; 1]$  stellt die normierte Kovarianz dar und wird wie folgt berechnet:

$$r(x, y) = \frac{n \sum xy - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2} \sqrt{n(\sum y^2) - (\sum y)^2}} \quad (4.12)$$

Der Funktionswert  $r(x, y)$  kann demnach wie folgt über  $n$  Stichproben interpretiert werden:

- +1: perfekter linearer Zusammenhang
- 0: gar kein linearer Zusammenhang
- -1: perfekter gegensätzlicher linearer Zusammenhang

Eine perfekte Korrelation von  $\pm 1$  tritt nur dann auf, wenn alle Datenpunkte auf einer Geraden liegen. Wobei der Anstieg dieser Geraden für  $r = +1$  positiv und für  $r = -1$  negativ ist. Im allgemeinen wird bei  $r > 0.8$  von einer starken Korrelation gesprochen und bei  $r < 0.5$  von einer schwachen.

### 4.3.5 Relief-Algorithmus für Mehrklassenanwendung

In [KR92] wurde der *Relief*-Algorithmus zur Merkmalselektion vorgestellt und als sehr effizient zur Bewertung der Merkmal-Relevanz eingestuft. Der Algorithmus generiert einen Gewichtsvektor über die Anzahl der Merkmale. Dieser wird anschließend bei einer sukzessiven Bewertung der Datensatz-Objekte aktualisiert, so dass das Gewichtsvektorelement eines relevanten Merkmals groß ist und für ein irrelevantes klein. Weiterhin werden die Parameter des Algorithmus analysiert mit dem Ziel, möglichst wenige irrelevante Merkmale auszuwählen. Der Algorithmus aus [GBNT04] zeigt die Vorgehensweise zur Bewertung der Merkmale, wobei  $nearhit(x)$  und  $nearmiss(x)$  den nächsten Punkt zu  $x$  in  $S$  (Datensatz) berechnen, mit der selben und der anderen Klasse. In [Kon94] wurde der ursprüngliche Algorithmus, der für zwei-Klassen-Probleme ausgelegt ist, auf Mehr-Klassen Probleme erweitert. Dabei wurden auch Verbesserungen des Algorithmus hinsichtlich verrauschter, fehlerbehafteter bzw. unvollständiger Merkmale vorgenommen.

Aus [WM04] geht hervor, dass erweiterte Formen des ursprünglichen Algorithmus, in diesem Fall der *Relief-F*, vergleichbare und zum Teil bessere Klassifikationsraten erreicht als andere Filter-Methoden. In [YFH<sup>+</sup>08] wird die Weiterentwicklung des ursprünglichen Algorithmus beschrieben und ebenfalls ein neuer Ansatz vorgestellt, der *Multi-Relief*. In [FYH<sup>+</sup>08] wird hervorgehoben, dass dieser besser sei als andere Methoden und sich damit hervorragend auch für Mehrklassen-Anwendungen eigne.

### 4.3.6 Informationsgewinn und Transinformation

Neben den linearen Methoden zur Bewertung verteilter Merkmale werden auch nichtlineare Methoden aus der Stochastik und Informationstheorie verwendet. Dabei werden die wesentlichen Informationen durch folgende Berechnungsvorschriften ermittelt:

- Wahrscheinlichkeit  $P$  für ein Merkmal  $m_i$  und einer Klasse  $fk_j$

$$P(m_i) = \frac{|Muster_{mit m_i}|}{|Muster_{gesamt}|} \quad (4.13)$$

$$P(\bar{m}_i) = 1 - P(m_i) \quad (4.14)$$

$$P(fk_j) = \frac{|Muster_{mitfk_j}|}{|Muster_{gesamt}|} \quad (4.15)$$

- Bedingte Wahrscheinlichkeit  $P$  eines Merkmals  $m_i$ , falls  $fk_j$  bereits eingetreten ist

$$P(m_i | fk_j) = \frac{|Muster_{ausfk_jmitm_i}|}{|Muster_{ausfk_j}|} \quad (4.16)$$

- Der Informationsgehalt  $I$  gibt an, wie viel Information in einer Nachricht  $X$  übertragen wurde.

$$I(X) = -\log_2(P(X)) \quad (4.17)$$

- Die Entropie  $H$  (Shannon, [SWS98]) beschreibt hingegen den mittleren Informationsgehalt bzw. den Erwartungswert des Informationsgehalts  $I(X)$ . Diese wird auch als Informationsdichte bezeichnet. Die Berechnung von  $H$  für eine Quelle  $X$  mit den Datensätzen  $x_i$  und den zugehörigen Wahrscheinlichkeiten  $p_i$  lässt sich nach folgender Gleichung berechnen.

$$H(X) = -\sum p_i \log_2(p_i) \quad (4.18)$$

- Die gemeinsame Entropie  $H(X, Y)$  beider unabhängiger Quellen  $X$  und  $Y$  mit den Datensätzen  $x_i$  und  $y_j$  wird wie folgt berechnet:

$$H(X, Y) = H(X) + H(Y) = -\sum P(x_i, y_j) \log_2(P(x_i, y_j)) \quad (4.19)$$

Bei Berücksichtigung der bedingten Wahrscheinlichkeit  $P(X | Y)$  des Ereignisses  $X$  unter der Bedingung  $Y$  spricht man auch vom bedingten Informationsgehalt und einer bedingten Entropie.

Die Wahrscheinlichkeitsberechnung und die Entropie sind Grundlage vieler Merkmalreduktionsverfahren, die beispielsweise auf dem Informationsgewinn, der Transinformation oder der symmetrischen Unbestimmtheit aufbauen. In den folgenden Abschnitten werden Verweise auf weiterführende Literatur gegeben und die Berechnungsvorschriften der ausgewählten Verfahren erläutert.

#### 4.3.6.1 Kullback-Leibler-Divergenz, Informationsgewinn

Die Kullback-Leibler-Divergenz wird auch als Informationsgewinn (*Information Gain*) IG oder relative Entropie bezeichnet und wird wie folgt definiert:

$$IG(Y | X) = \sum P(x_i) \log \left( \frac{P(x_i)}{P(y_j)} \right) \quad (4.20)$$

Dieses nicht-symmetrische Maß dient zur Bewertung der Unterschiedlichkeit zweier Wahrscheinlichkeitsverteilungen  $X$  und  $Y$  des selben Ereignisses. In [Aya99] wird die Verwendung des Informationsgewinns zur Gewichtung der zur Klassifikation verwendeten Merkmale erläutert. Die Inspiration zur Umsetzung basiert auf den Veröffentlichungen von [Qui86, Qui93]. Es wurde darüber hinaus der Effekt auf die Klassifikationsrate für verschiedene durch den Informationsgewinn gewichtete Merkmale (zum Teil auch durch künstlich erzeugte und hinzugefügte Merkmale) untersucht. Es wurde gezeigt, dass diese Methodik für einige untersuchte Datensätze eine Verbesserung von bis zu 15% erreicht.

#### 4.3.6.2 Transinformation, gegenseitige Information

Die Transinformation wird auch als *Mutual Information*  $MI$  oder gegenseitige Information bezeichnet. Diese dient als Maß für die Stärke des statistischen Zusammenhangs zweier Zufallsgrößen und steht in einem engen Zusammenhang zur Entropie beziehungsweise zur bedingten Entropie nach [Sha48]. Die Distanz zwischen zwei Wahrscheinlichkeitsverteilungen kann durch den Informationsgewinn bzw. relative Entropie bestimmt werden, Gleichung 4.20.

Die *Mutual Information* wird nun über den  $IG$  zwischen der Wahrscheinlichkeit zweier Variablen  $P(X, Y)$  und dem Produkt ihrer einzelnen Wahrscheinlichkeiten  $P(X) P(Y)$  berechnet. Die gemeinsame Wahrscheinlichkeit  $P(X, Y)$  (Verbundwahrscheinlichkeit) ist dabei nicht mit der bedingten Wahrscheinlichkeit  $P(X | Y)$  zu verwechseln.

$$MI(X; Y) = \sum_x \sum_y P(x_i, y_j) \log_2 \frac{P(x_i, y_j)}{P(x_i) P(y_j)} \quad (4.21)$$

Die Transinformation kann auch über die Differenz der Quell-Entropie (Entropie  $H(X)$ ) und der Fehlinformation (bedingte Entropie  $H(X | Y)$ ) definiert werden. Weiterhin ist die  $MI$  symmetrisch, so dass gilt:

$$MI(X; Y) = MI(Y; X) = H(X) - H(X | Y) = H(Y) - H(Y | X) \quad (4.22)$$

$MI$  beschreibt den Informationsgehalt, über den die beiden Wahrscheinlichkeitsverteilungen wechselseitig verfügen. Für Anwendungen im Bereich der Merkmalreduktion wird  $MI(m_i, fk_j)$ , bezogen auf  $m_i$  Merkmale und  $fk_j$  Klassen, berechnet, und zur Selektion eines Merkmals  $m_i$  werden beispielsweise die  $MI$  Werte für alle Klassen  $fk_j$  gewichtet nach deren Wahrscheinlichkeit  $P(fk_j)$  summiert:

$$MI(m_i) = \sum_{j=1}^k P(fk_j) MI(m_i; fk_j) \quad (4.23)$$

Für die Entscheidung zwischen  $k$  Klassen werden anschließend die Merkmale  $m_i$  ausgewählt, für die  $MI(m_i)$  am größten ist. Weiterführende Informationen zu Selektionsalgorithmen, wie zum Beispiel mRMR (*minimum Redundancy Maximum Relevance Feature Selection*), sind in [DP03, DP05, PLD05] gegeben.



#### 4.3.6.3 Symmetric-Uncertainty/ Symmetrische Unbestimmtheit und Redundanz

Bei der symmetrischen Unbestimmtheit handelt es sich um die normalisierte Variante der Transinformation. Im ersten Schritt wird deshalb anhand der *Mutual Information* nach Gleichung 4.21 ein Wert für die Redundanz zwischen den Quellen  $X$  und  $Y$  berechnet. Die Redundanz  $R$  bestimmt den Informationsgehalt auf Basis der  $MI$ , normiert durch die Entropie der Quelle  $X$  und  $Y$ .

$$R(X;Y) = \frac{MI(X;Y)}{H(X) + H(Y)} \quad (4.24)$$

Nach [WF05] wird die symmetrische Unbestimmtheit (*Symmetric-Uncertainty*) wie folgt aus der Redundanz berechnet:

$$SU(X;Y) = 2 \cdot \frac{MI(X;Y)}{H(X) + H(Y)} = 2 \cdot R(X;Y) \quad (4.25)$$

Die  $SU$  ist eine modifizierte Variante der  $MI$ , normiert diese in das Intervall  $[0,1]$  und wird zur Bestimmung der nichtlinearen Korrelation zwischen zwei Variablen angewandt.

#### 4.3.7 Fast Correlation Based Filter - FCBF

Das in [YLG04] vorgestellte Verfahren zur Merkmalreduktion basiert auf der Korrelationsanalyse des Datensatzes. Abgesehen von der Relevanz wird dadurch auch die Redundanz beurteilt und mit bei der Reduktion des Datensatzes berücksichtigt. Bei der Verwendung der linearen Korrelationsanalyse, siehe Unterabschnitt 4.3.4, können nichtlineare Zusammenhänge in den Datensätzen nicht erfasst werden, weshalb das *FCBF* Verfahren auf der aus der Informationstheorie bekannten Entropie, ein Maß für die Unbestimmtheit einer zufällig ausgewählten Variable, aufbaut. Die Definition des Informationsgehalts (Entropie) ist in Abschnitt 4.3.6 gegeben. Durch die bedingte Entropie kann der Informationsgehalt für eine Quelle  $X$  unter der Bedingung  $Y$  mit den Datensätzen  $x_i$  und  $y_j$  ermittelt werden:

$$H(X|Y) = - \sum_j P(y_j) \sum_i P(x_i | y_j) \log_2(P(x_i | y_j)) \quad (4.26)$$

Wird die bedingte Entropie der Quelle  $X$  unter der Bedingung  $Y$  von der Entropie der Quelle  $X$  subtrahiert so erhält man die Transinformation (fälschlicher Weise wird in [YLG04] von dem Informationsgewinn gesprochen, siehe Gleichung 4.21 und 4.22 bzw. Unterabschnitt 4.3.6.2). Als Beispiel sei die Bedingung  $MI(X;Y) > MI(Z;Y)$  genannt, welche aussagt, dass Quelle  $Y$  eine höhere Korrelation zu Quelle  $X$  aufweist als zu Quelle  $Z$ . Die *Mutual Information* ist symmetrisch. Um den Einfluss der Quellenmenge und daraus eventuell resultierender Ungleichgewichtung bei der Berechnung der  $MI$  für zwei Quellen ungleichen Umfangs zu vermeiden, wird die normalisierte Variante der Transinformation, die symmetrische Unbestimmtheit (*Symmetric Uncertainty*) nach Gleichung 4.25 verwendet.

Title	FCBF <sub>(log)</sub>	FCBF <sub>(0)</sub>		Full Set		ReliefF		CFS-SF		FOCUS-SF	
	Acc	Acc	p-Val	Acc	p-Val	Acc	p-Val	Acc	p-Val	Acc	p-Val
Lung-cancer	83.33	86.67	0.34	78.33	0.34	84.17	0.85	86.67	0.34	87.5	0.46
Promoters	93.27	93.27	1	91.55	0.55	87.82	0.25	95.18	0.17	90.45	0.40
Splice	93.95	96.14	<b>0.00</b> <sup>+</sup>	95.52	<b>0.00</b> <sup>+</sup>	91.32	<b>0.00</b> <sup>-</sup>	93.54	0.24	94.36	<b>0.08</b> <sup>+</sup>
USCensus90	97.94	97.88	0.19	93.49	<b>0.00</b> <sup>-</sup>	97.97	0.17	97.99	0.65	97.87	0.44
CoIL2000	93.94	93.92	0.34	78.68	<b>0.00</b> <sup>-</sup>	93.89	0.66	92.92	<b>0.01</b> <sup>-</sup>	83.22	<b>0.00</b> <sup>-</sup>
Chemical	71.91	67.73	<b>0.02</b> <sup>-</sup>	60.90	<b>0.00</b> <sup>-</sup>	71.26	0.77	70.51	0.35	66.35	<b>0.00</b> <sup>-</sup>
Musk2	84.59	84.59	1	84.78	0.51	84.59	1	64.87	<b>0.00</b> <sup>-</sup>	83.53	<b>0.01</b> <sup>-</sup>
Arrhythmia	67.48	65.73	0.45	60.88	<b>0.01</b> <sup>-</sup>	55.79	<b>0.00</b> <sup>-</sup>	69.05	0.45	69.06	0.56
Isolet	50.06	83.33	<b>0.00</b> <sup>+</sup>	84.10	<b>0.00</b> <sup>+</sup>	60.90	<b>0.00</b> <sup>+</sup>	87.31	<b>0.00</b> <sup>+</sup>	71.03	<b>0.00</b> <sup>+</sup>
Multi-feat	95.9	95.65	0.50	94.1	<b>0.01</b> <sup>-</sup>	67.65	<b>0.00</b> <sup>-</sup>	96.15	0.64	93.7	<b>0.02</b> <sup>-</sup>
L/W/T	-	1/2/7		5/2/3		3/1/6		2/1/7		4/2/4	

Abbildung 4.3: Vergleich des *FCBF*-Algorithmus mit anderen Reduktionsverfahren, aus [YLG04].

Angenommen, die Werte der einen Quelle enthalten die Informationen der anderen Quelle, so würde sich dies in einem Wert von 1 bei der Berechnung der *SU* äußern. Bei einem Wert von 0 sind die beiden Quellen voneinander unabhängig. Die *SU* wird in dem *FCBF*-Algorithmus als Maß für die Korrelation zwischen zwei Variablen angewendet. Dabei wird laut [YLG04] unterschieden in:

- *C-correlation*:  
Steht als Maß für die Korrelation zwischen jedem Merkmal  $F(i)$  und der Klasseninformation  $C$ , dargestellt durch  $SU(F(i), C)$
- *F-correlation*:  
Steht als Maß für die Korrelation zwischen jedem Paar der Merkmale  $F_i$  und  $F_j$  ( $i \neq j$ ), dargestellt durch  $SU(i, j)$

Betrachtet man den Aufwand zur Berechnung der *F-correlation*, wo  $\frac{N(N-1)}{2}$  Merkmalpaare gebildet werden müssen, so lässt sich daraus ableiten, dass dieser Algorithmus vor allem für niedrig dimensionale Datensätze geeignet ist. Durch die Berechnung und Bewertung der *SU* für die möglichen Merkmal- und Klassenkombinationen ( $SU(F(j), C)$ ;  $SU(F(i), C)$ ;  $SU(F(i), F(j))$ ) wird eine Markovdecke (*Markov blanket*) gebildet, welche zur Bestimmung der besten Merkmale genutzt wird.

Verallgemeinert besteht der *FCBF*-Algorithmus aus zwei Phasen. Die erste Phase ist die Definition der relevanten Merkmale (Zeile 2-7), und die zweite Phase berechnet die besten (*predominant*; vorherrschend) Merkmale aus den relevanten Merkmalen (Zeile 8-20). Zusammengefasst, erfolgt die Auswahl erst über die *SU*, das Maß für die Korrelation zwischen den Merkmalen und den Klassen  $SU(F(i), C)$  und anschließend über die Redundanz zwischen den Merkmalen, welche durch  $SU(F(i), F(j))$  berechnet wird. Weiterführende ausführliche Beschreibungen des *FCBF*-Algorithmus' sind in [YLG04] gegeben.

Es zeigt sich in Abbildung 4.3 am Beispiel einiger Datensätze aus der UCI Bibliothek [AN07], dass der *FCBF* für den Datensatz CoIL2000 durchgängig die besten Ergebnisse liefert (im Vergleich mit drei anderen Reduktionsverfahren). Bei dem Lung-Cancer oder Splice Datensatz hingegen erreichen alle Verfahren vergleichbare Erkennungsraten. *Acc* bezeichnet laut [YLG04] die Erkennungsraten bei einer 10-fachen Kreuzvalidierung in % und *p-Val* den Wahrscheinlichkeitswert eines t-Tests.

## 4.4 Mustererkennung zur Fehlergeräuscherkennung

Bei der Klassifizierung von Mustern wird verallgemeinert auch von der Mustererkennung gesprochen. Dabei werden durch ein Klassifikationsverfahren die Muster des diskreten, meist mehrdimensionalen Merkmalraums Klassen zugeordnet. Da die Klassifizierung anwendungsbezogen ist, existieren diverse Verfahren mit unterschiedlichen Beschreibungsformen (siehe Abbildung 4.4).

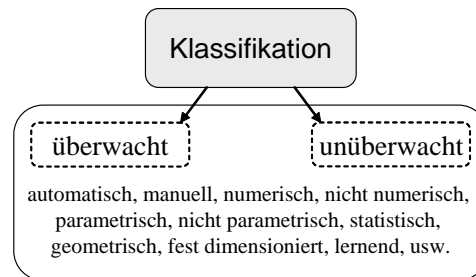


Abbildung 4.4: Beschreibungsformen von Klassifikationsverfahren

Prinzipiell können diese in zwei Bereiche eingeteilt werden:

- **Überwachte Verfahren - Lernen mit Unterweisung:**

Anhand von Stichproben/ Mustern wird die Abweichung der ermittelten Klasseninformation von der tatsächlichen Klasse ermittelt und dadurch der Lernvorgang beeinflusst. Man spricht in diesem Fall auch von manueller Klassifizierung oder *supervised learning*, da der Mensch die zu klassifizierenden Muster mit einer Klasseninformation versieht. Zu jeder Eingabe besteht demzufolge eine korrekte Ausgabe, aus der sich ein Korrekturfaktor für die Lernregel<sup>3</sup> des Klassifikators ergibt.

- **Unüberwachte Verfahren - Lernen ohne Unterweisung:**

Die zu klassifizierenden Muster werden selbstständig mit Klasseninformationen versehen, ohne Zuhilfenahme von bereits kategorisierten Mustern. Meist ist das Ziel bei der Anwendung solcher Verfahren maschinelle/ technische Systeme automatisch und ohne menschliche Hilfe klassifizieren zu lassen. Das Lernen findet unüberwacht (*self-organized learning*), also ohne Vorgabe eines korrekten Ausgabewertes statt. Es existieren diverse Lernregeln, die je nach Klassifikatortyp angepasst sind (Hebbsche Lernregel, SOM, *k-means Cluster Analysis*, Vektorquantisierung bzw. *Vector Quantization*, usw.).

- **Bestärkendes Lernen:**

Stellt eine Mischform dar, bei der kein korrekter Ausgabewert vorgegeben wird, sondern die Lernregel lediglich anhand von Belohnung und Bestrafung die Anpassung der Klassifikator-Gewichte ableitet.

Für die Realisierung eines Agenten zur automatischen Erkennung von Geräuschen sollen sowohl selbstlernende als auch überwachte Klassifikatoren in kaskadierter Form (Abbildung 4.5) verwendet werden. Dabei wird bei einer nicht in Ordnung (n.i.O.) Klassifikation durch ein unüberwachtes

<sup>3</sup>Die Lernregel verändert die Klassifikator-Gewichte so, dass der Fehler kleiner wird.

Mustererkennungsverfahren eine detaillierte Mustererkennung durch ein überwachtes Verfahren angewandt. Dieses bestimmt die genaue Fehlerklasse  $fk$  des Merkmalvektors  $m$ .

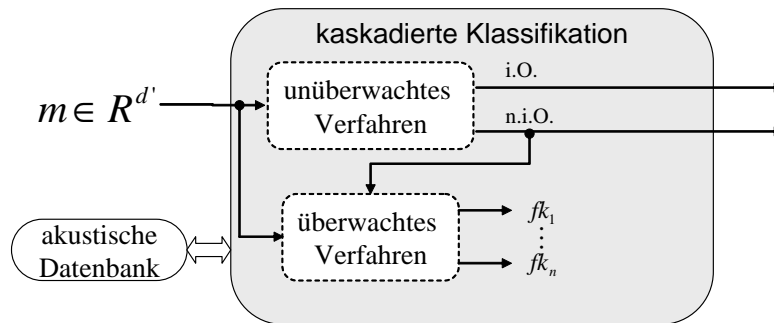


Abbildung 4.5: Prinzip der kaskadierten Klassifikation des Merkmalvektors  $m$

Der Entwurf und die Validierung der geforderten Funktionen und Klassifikationsraten (siehe Gleichung 3.1) soll unter Matlab® erfolgen. Dennoch soll auch untersucht werden, in wie weit Algorithmen für Steuergeräteintegration, mit hohen Anforderungen an Recheneffizienz, verwendet werden können. Die Klassifikationsmethoden benötigen eine Grundmenge an Lerndaten (Trainingsdatensatz), um deren Struktur bzw. Muster zu lernen oder zu erkennen. Der Test der Klassifikationsverfahren erfolgt anhand eines Testdatensatzes. Je nach Klassifikationsaufgabe müssen folgende Herausforderungen bewältigt werden:

- Ausreißer, die beispielsweise durch Mess- oder Berechnungsfehler hervorgerufen werden
- Eindeutige Merkmale für die definierten Klassen (z.B. bilden dröhnende Geräusche eine Klasse und brummende die andere) sind teilweise schwierig zu definieren
- Falsche Merkmale haben keine Aussagekraft über die Klassenzugehörigkeit. Die Farbe eines Fahrzeugs stellt ein falsches Merkmal dar, wenn die Geräusche klassifiziert werden sollen.
- Fließende Übergänge zwischen den Klassen sollten vermieden werden.
- Nichttrennbarkeit der Klassen liegt meist dann vor, wenn entweder zu wenige oder klassenunabhängige Merkmale vorliegen.

In den folgenden Abschnitten dieses Kapitels werden verschiedene Verfahren zur Mustererkennung näher erläutert. Insbesondere wird dabei auf selbstorganisierende Karten eingegangen.

### 4.4.1 Unterteilung der Datenmenge in Test-/ Training-/ Validierungsmenge

Es existieren verschiedene Methoden zur Vorbereitung und Einteilung der zur Verfügung stehenden Datensätze, um auch bei nur wenigen Lerndaten vernünftig klassifizieren zu können. Die hier vorgestellten Methoden sind ausführlich in [WF05] beschrieben.

#### • **split-sample- und hold-out-Methode**

Unter diesen Begriffen wird die Einteilung des zur Verfügung stehenden Datensatzes in einen Trainingsdatensatz, einen Testdatensatz und wahlweise einen Validierungsdatsatz durch eine vorgegebene prozentuale Einteilung bezeichnet. Dabei ist zu beachten, dass für eine

möglichst große Repräsentativität möglichst alle Klassen in den Teilmengen vorkommen (Stratifikation: jede Klasse kommt mit annähernd gleicher relativer Häufigkeit in den Teilmengen vor, siehe Abbildung 4.6). Ein Beispieldatensatz mit 300 Mustern wird hier so aufgeteilt, dass ein Testdatensatz mit 75 Mustern und ein Trainingsdatensatz mit 225 Mustern für einen Klassifikatortest zur Verfügung steht. Zum Testen wird in [WF05] ein Drittel, zum Training zwei Drittel empfohlen. Der Validierungsdatensatz wird wahlweise genutzt, um beispielsweise ein Abbruchkriterium zu definieren oder Lernparameter zu optimieren. Durch wiederholte Generierung zufällig erzeugter Teilstichproben kann der *hold-out* Prozess zuverlässiger gemacht werden (*repeated hold-out*). Dabei können jedoch überschneidende Teilmengen entstehen. Die Klassifikationsrate des gesamten Datensatzes wird durch Mittelung berechnet.

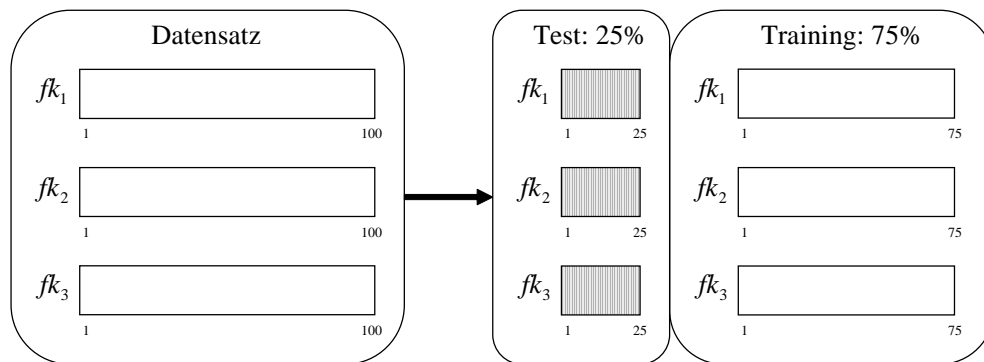


Abbildung 4.6: Die Generierung eines Trainings- und Testdatensatzes anhand der *hold-out* Methode

#### • Kreuzvalidierung

Diese Methode, auch *n-fold cross-validation* genannt, vermeidet die Generierung überschneidender Teilmengen. Dafür wird der Datensatz in  $n$  Teilmengen gleicher prozentualer Größe zerlegt (siehe Abbildung 4.7) und in einem Test- und Trainingsdatensatz zusammengeführt. Anschließend werden diese aus den nächsten Teilmengen gebildet. Der Prozess wird so lange durchgeführt, bis jede aus dem Datensatz extrahierbare Teilmenge genutzt wurde. Die Klassifikationsrate des gesamten Datensatzes wird durch Mittelung berechnet. In [WF05] wird eine 10-fache Kreuzvalidierung empfohlen (anhand theoretischer Begründungen und Experimenten gezeigt), um eine zuverlässige Schätzung zu erhalten. Einen Sonderfall stellt die *leave-one-out cross-validation* dar, bei der jeweils nur ein Muster den Testdatensatz darstellt und alle anderen Muster den Trainingsdatensatz bilden. Jede Klassifikationsrate der Testdatensätze ist nur für eine Klasse repräsentativ, nach der Mittelung jedoch für alle Klassen. Dieses Verfahren bewertet somit jedes Muster unabhängig, benötigt dafür jedoch einen großen Rechenaufwand.

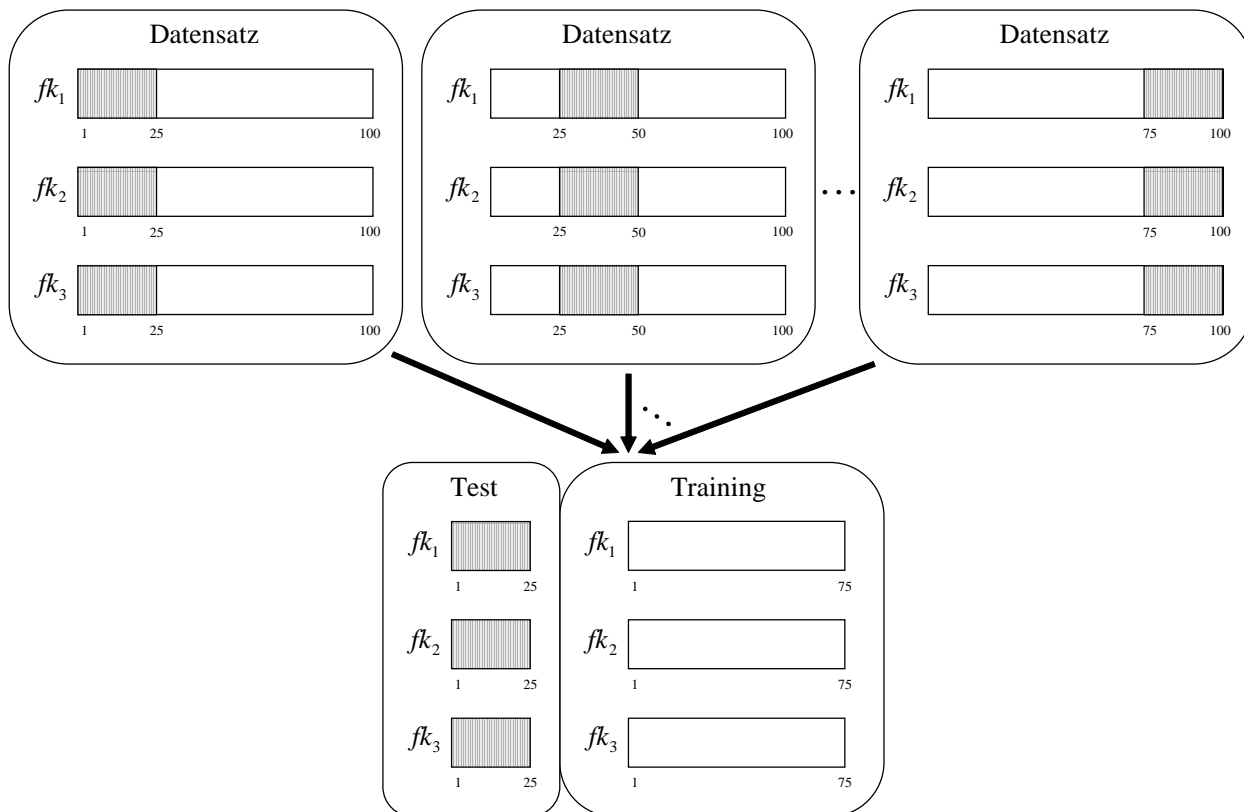


Abbildung 4.7: Kreuzvalidierungsverfahren am Beispiel eines Datensatzes mit 3 Fehlerklassen und 300 Mustern

### • *bootstrap*-Methode

Im Gegensatz zur Kreuzvalidierung wird hier über eine Anzahl von  $n$  Iterationsschritten ein zufällig ausgewähltes Muster aus dem Datensatz extrahiert und in einem Testdatensatz zusammengeführt. Die verbleibenden Muster des Datensatzes werden als Trainingsdatensatz verwendet. Bezogen auf einen Beispieldatensatz mit 300 Mustern würden 100 Muster zufällig extrahiert als Testdaten und 200 Muster zum Anlernen verwendet werden. Durch wiederholte zufällige Generierung eines Test- und Trainingsdatensatzes wird die mittlere Klassifikationsrate berechnet. Diese Methode eignet sich laut [WF05] vor allem für kleine Datenmengen.

### 4.4.2 Distanzmaße zur statistischen Mustererkennung

Im Bereich der statistischen Mustererkennung werden Distanzmaße zur Untersuchung des Merkmalraums und Identifikation der Klassen genutzt. Dazu werden für den zu untersuchenden Vorgang im ersten Schritt Merkmale bestimmt. Danach folgt das Auftragen im Merkmalraum. Statistische Methoden zur Bestimmung von Trennflächen werden abschließend für die Identifikation der Klassen genutzt. In Abbildung 4.8 ist an einem Beispiel verdeutlicht, wie die Muster zweier Merkmale einen Merkmalraum aufspannen. Es zeigt sich, dass eine grafische Auswertung nur bis 3 Merkmale möglich ist. Die Dimension des Merkmalraums wird jedoch durch die Anzahl der Merkmale bestimmt,

welche den meist hochdimensionalen Raum aufspannen. Durch Kombination und Auswahl der Merkmale wird eine Reduktion der Dimension erreicht - siehe Erläuterungen zur Merkmalselektion in Abschnitt 4.3. Die Merkmale werden so gewählt, dass zwischen den Klassen unterschieden werden kann und sich, wenn möglich, Punktwolken mit definierten klassenabhängigen Zentroiden (*Cluster*) bilden. In den meisten Fällen sind die *Cluster* selten kompakt und trennbar. Die Bewertung erfolgt durch die Distanzmaße [Lam05], dabei soll die Distanz zu jedem Muster der selben Klasse kleiner sein als die Distanz zum nächsten Nachbarn jeder anderen Klasse. Die Berechnung der Distanz kann durch verschiedene Berechnungsmethoden, die in diesem Abschnitt vorgestellt werden, erfolgen. Diese Abstandsmaße werden sowohl zur Generierung eines reduzierten Merkmalraums (Merkmalreduktionsalgorithmen) als auch zur Bestimmung der wahrscheinlichen Klassen (Mustererkennungsalgorithmen) genutzt.

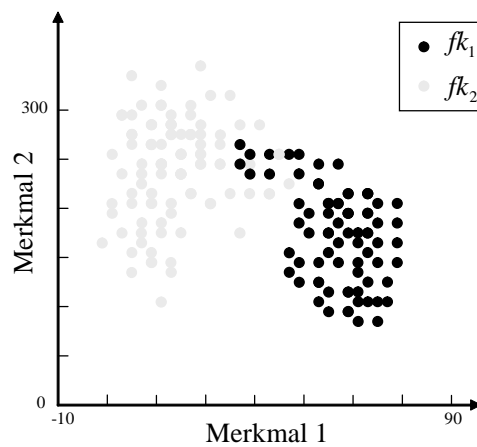


Abbildung 4.8: Merkmalraum; Muster gleicher Klasse sind farblich markiert (grau =  $f_{k_1}$ ; schwarz =  $f_{k_2}$ )

Für folgende Gleichungen gilt:

$d_{i,j}$ : Distanz der Muster  $i$  und  $j$  (bestehen aus  $k$ -Merkmalen),

$m_{ik}, m_{jk}$ : Wert des Merkmals  $k$  bei Muster  $i, j$ ,

$K$ : Anzahl der Merkmale.

#### • Minkowski-Metriken, L1 und L2 Norm

Für viele Algorithmen werden die Minkowski-Metriken oder L-Normen verwendet. Die wohl bekannteste stellt die Euklidische Distanz dar. Die Minkowski-Konstante  $r \geq 1$  definiert die L1- und L2-Norm. Die Berechnung erfolgt durch folgende Gleichung:

$$d_{i,j} = \left[ \sum_{k=1}^K |m_{ik} - m_{jk}|^r \right]^{\frac{1}{r}} \quad (4.27)$$

$r = 1$  Manhattan- oder Cityblock-Metrik (L1-Norm)

$r = 2$  Euklidische Distanz (L2-Norm)

- **Mahalanobis- und robuste Distanz**

Die Mahalanobis-Distanz [Mah36] gibt für jeden Datenpunkt an, wie weit dieser vom Zentrum der Daten (unter Berücksichtigung der Streuung) entfernt ist. Diese kann anstelle der Euklidischen Distanz verwendet werden (z. B. in neuronalen Netzen), vorwiegend wird diese jedoch zur Detektion von Ausreißern verwendet. Die Muster einer Klasse bilden eine Punktwolke/*Cluster*. Ein Muster, welches stark außerhalb liegt wird als Ausreißer bezeichnet. Der Abstand zwischen zwei Mustern  $m_i$  und  $m_j$  wird wie folgt berechnet:

$$d_{i,j} = \sqrt{(m_i - m_j)^T C(X)^{-1} (m_i - m_j)} \quad (4.28)$$

$C(X)$ : Kovarianzmatrix

Wobei bei einer Ausreißerbestimmung beispielsweise  $m_j$  durch das komponentenweise arithmetische Mittel aller Muster ersetzt würde. Die Kovarianzmatrix, wie auch das arithmetische Mittel werden stark durch Ausreißer beeinflusst. Über eine robuste Schätzung dieser Größen kann entsprechend der gegebenen Gleichung eine robuste Distanz berechnet werden.

- **Maximum-Metrik**

Bei der Maximum-Metrik wird die größte Distanz zwischen  $K$  Merkmalen der Muster  $i$  und  $j$  berechnet.

$$d_{i,j} = \max_{k=1 \dots K} |m_{ik} - m_{jk}| \quad (4.29)$$

- **Canberra-Metrik**

Die Canberra-Metrik stellt eine normierte Metrik dar, bei der es keine Distanzen über 1 gibt. Sie wird wie folgt berechnet:

$$d_{i,j} = \frac{1}{K} \sum_{k=1}^K \frac{|m_{ik} - m_{jk}|}{|m_{ik} + m_{jk}|} \quad (4.30)$$

#### 4.4.3 Klassifikatoren auf neuronaler Basis

Künstliche neuronale Netze (KNN, *artificial neural network*, ANN) sind Modelle, die basierend auf biologischen Erkenntnissen ein vereinfachtes Abbild eines menschlichen Gehirns darstellen. KNN stellen ein informationsverarbeitendes System dar, welches aus mehreren Neuronen besteht. Der Austausch der Information erfolgt über Synapsen (gewichtete Verbindungen). Diese können den Informationsaustausch der Neuronen entweder verstärken oder dämpfen. Ziel des realisierten neuronalen Netzes ist es, aus einem vorliegenden Beispiel zu lernen und dieses Wissen auf unbekannte Eingaben anzuwenden. Dadurch sollen besonders schwierig zu lösende Probleme der Signal- und Informationsverarbeitung bearbeitet werden, die mit anderen Methoden der klassischen Mathematik, Statistik oder Informationstheorie nur unzureichend oder gar nicht gelöst werden können. Als Beispiel dafür soll die Schrifterkennung genannt sein, bei der Buchstaben und Zahlen



durch jeden Menschen leicht oder stark unterschiedlich geschrieben werden und dadurch eine besondere Herausforderung an einen Schrifterkennungsalgorithmus stellen. Trotz dieser großen Varianz soll anhand einer begrenzten Auswahl an Beispielen möglichst alle Buchstaben und Ziffern erkannt werden, siehe [MGS05, PC95].

In den letzten Jahren hat sich ein breites Anwendungsgebiet für beispielsweise physikalische, biologische und informatische Problemstellungen durch KNN erschlossen. In diesem Abschnitt sollen der Aufbau, die Struktur und die Lernalgorithmen vorgestellt werden. Dies soll jedoch nur auf wesentliche Punkte eingehen, da weitläufig detaillierte Informationen zur Theorie ([MST94, BJGS01] und [Hay94]) und Anwendungen ([Sch07, RBK96, MD96]) in verschiedenster Literatur gegeben sind.

#### 4.4.3.1 Neuronenmodelle und Netztopologien

Die Unterscheidung der KNN erfolgt hauptsächlich anhand ihren Neuronenmodellen und der Netztopologie.

- **McCulloch und Pitts Neuronenmodell nach [MP43]**

Verfügt über erregende und hemmende Eingänge. Der Ausgang wird gesetzt, wenn die Summe der erregenden Eingänge (binäre Signale) einen Schwellenwert überschreitet und keine hemmender Eingang gesetzt ist. Dieses Neuronenmodell ist nur eingeschränkt dem biologischen Vorbild entsprechend und nutzbar. Das heutige Neuronenmodell stellt eine Weiterentwicklung des Modells von [MP43] dar.

- **Perceptron Neuronenmodell nach [Ros58]**

Stellt bis heute die Grundlage künstlicher neuronaler Netze dar. Das einlagige Perceptron wird durch nur eine Schicht von künstlichen Neuronen gebildet. Diese bestehen aus einem Eingabevektor und Gewichtungen, einer Übertragungs- und Aktivierungsfunktion und einer Ausgabe. Das Lernen eines Perzeptrons erfolgt durch die Anpassung der Gewichtung eines jeden Neurons.

- **Multi-Layer-Perceptron Neuronenmodell**

Stellt ein mehrlagiges Perzeptron dar, mit einer Eingabe- und Ausgabeschicht. Dazwischen liegen verdeckte Schichten (*hidden layer*). Die Neuronen einer Schicht sind vollständig mit den Neuronen der folgenden Schicht durch wahlweise veränderbare Gewichte verbunden (*Feedforward-Netz*). Durch die mehrschichtige Auslegung basierend auf linearen Perzeptrons und nichtlineare Aktivierungsfunktionen können auch Vorgänge, die nicht linear trennbare Daten aufweisen, differenziert werden. Mehrschichtige Perzeptronen-Netze (*Multi-Layer Perceptron Neural Network*, MLP-NN) sind dadurch leistungsfähiger.

- **Vorwärtsgerichtete Topologie**

Dabei bestehen die *Feedforward*-Netze aus mehreren Schichten und die Verbindungen bestehen nur von einer Schicht zur nächsten. Man nennt diese Topologieform auch vollständig verbundene Netze (z.B. MLP). Einen Sonderfall stellen allgemeine vorwärtsgerichtete Netze mit Verbindungen, die Schichten überspringen, dar (*shortcut connections*).

- **Netztopologie mit Rückkopplung**

Rekurrente-KNN (*recurrent-ANN*) verfügen über Verbindungen zwischen allen Schichten mit nicht nur vorwärts gerichtetem Informationsaustausch. Dabei wird unterschieden zwischen neuronalen Netzen mit:

- direkten Rückkopplungen (*direct feedback*),
- indirekten Rückkopplungen (*indirect feedback*),
- seitlichen Rückkopplungen (*lateral feedback*, z.B. *Winner-Takes-All*-Netze wie SOM oder LVQ nach [Koh01]),
- vollständigen Verbindungen (z.B. Hopfield-Netze nach [Hop82]).

### 4.4.3.2 Wichtige Lernalgorithmen für Feedforward-Netze

- **Backpropagation Algorithmus nach [RHW88]**

Dieser zählt zu den bekanntesten Algorithmen. Es erfolgt eine Anpassung der Gewichte durch ein Gradienten-Abstiegsverfahren:

1. Ein Trainingsmuster wird dem KNN präsentiert.
2. Vergleich der KNN-Ausgabe mit der Klasseninformation des Trainingsmusters. Berechnung des Fehlers in jedem Ausgabeneuron.
3. Berechnung des lokalen Fehlers für jedes Neuron. Über die Differenz des Neuron-Ausgabewertes zu der Vorgabe wird ein Skalierungsfaktor/ Fehlergradient berechnet. Dieser sagt aus, um wie viel der Ausgabewert verstärkt oder gedämpft werden muss, um die Vorgabe zu erreichen.
4. Anpassung der Neuronengewichte zur Verringerung des lokalen Fehlers.
5. Generierung eines Rückgabewerts, korrespondierend zum lokalen Fehler, anhand dessen die Anpassung der Gewichte in den versteckten Schichten erfolgt.
6. Wiederholung ab Punkt 3. für die Neuronen jeder unterliegenden versteckten Neuronenschicht.
7. Wiederholung ab Punkt 2.

Es existiert eine Vielzahl von angepassten Algorithmen, bei denen beispielsweise eine variable Lernrate zur Verhinderung von oszillierenden Netzen oder ein Trägheitsterm zur Beschleunigung des Adaptionprozesses eingesetzt wird [Leu02].

- **Delta-Regel**

Der Backpropagation-Algorithmus stellt eine Verallgemeinerung der Delta-Regel (Perzeptron-Lernregel), auf mehrschichtige Netze angewandt, dar. Man bezeichnet die Delta-Regel der Neuroinformatik auch als LMS-Algorithmus in der Signalverarbeitung (*Least-Mean-Squares*-Algorithmus). Es beruht auf dem Gradientenverfahren (Verfahren des steilsten Abstiegs). Bei

diesem Verfahren - für einschichtige Netze - muss eine externe Vorgabe existieren, die den Zielwert für jede Eingabe definiert. Die Berechnung erfolgt durch folgende Formel:

$$\Delta w_{ij} = \eta \delta_j o_j \quad (4.31)$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (4.32)$$

$w_{ij}$ : Neuronengewichte

$\eta$ : Lernrate

$\delta_j$ : Fehlerwert,  $\delta_j = \text{soll}_j - \text{ist}_j = t_j - o_j$

- **Super-SAB nach [Tol90]**

Stellt eine veränderte Variante des Backpropagation-Algorithmus' dar. Die hauptsächliche Änderung besteht in der Einführung von gewichtsabhängigen Lernraten, da davon ausgegangen wird, dass jedes Gewicht des Netzes eine andere Fehlerfunktion aufweist. Aus [Leu02] geht hervor, dass dieser Algorithmus schneller konvergiert als das einfache Gradientenverfahren. Ein Vergleich zu anderen Lernalgorithmen ist in [Rie94] gegeben.

#### 4.4.4 Self-Organizing-Maps

Die SOM ist ein neuronales Netz (auch Kohonen-Netz oder -Karte genannt, siehe [Koh01]), welches sich sehr stark an der Funktionsweise des Gehirns orientiert. Die Neuronen werden durch einen mehrdimensionalen Gewichtsvektor repräsentiert. Bei einem adaptierten Neuronennetz haben sich die Gewichtsvektoren an den Eingaberaum selbstständig (selbstorganisierend bzw. ohne Lehrer) angepasst. Aus der Biomedizintechnik geht hervor, dass für jede Handlung/ Funktion des Gehirns unterschiedliche Bereiche genutzt werden. Durch bildgebende Verfahren werden bei entsprechender Sinnesanregung die aktiven Bereiche des Gehirns identifiziert. Dadurch kann z.B. ermittelt werden, in welcher Region im Gehirn die Erinnerung an ein bestimmtes Ereignis vorliegt. Visionen der Mediziner gehen bei diesen Methoden sogar in die Bereiche des Gedankenlesens oder der Lügendetektion. Versteht man die SOM als einfache Nachbildung eines Gehirns, dann kann diese Form des Neuronalen Netzes demzufolge als Wissensspeicher genutzt werden. Eine Übersicht für die Anwendungsbereiche in verschiedensten Gebieten zeigt [KK96]. Hauptanwendungsgebiet ist die Erkennung von Clustern in hochdimensionalen Datenmengen. Demzufolge wird erwartet, dass bei verschiedenartigen Anregungen jeweils andere Bereiche des Neuronennetzes aktiv sind. Durch den so genannten „labeling“ Prozess werden die aktiven Neuronenbereiche der Anregung zugeordnet.

Weitere Informationen zu Aufbau und Anwendung von SOM's sind in [Ste06, Kan04, SG04, Le05] gegeben. In diesem Abschnitt soll sehr detailliert ein Einblick in die Funktionsweise des SOM-Algorithmus' und deren Anwendung eingegangen werden, da dieser die Grundlage für den Agent zur Fehlerindikation in Kapitel 6 darstellt.

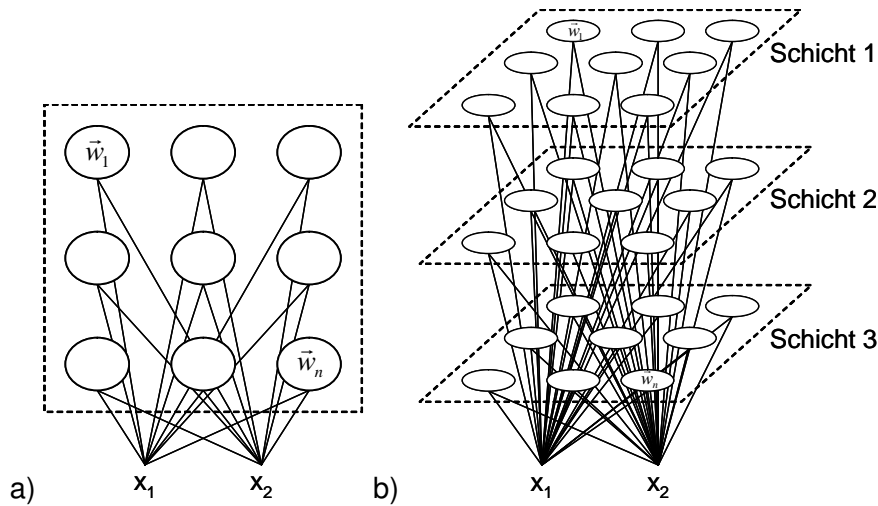


Abbildung 4.9: a) 2-dimensionales Kohonen-Karten-Modell mit 9 Neuronen b) 3-dimensionale Anordnung der Neuronen

Jeder Eingabeknoten besitzt eine Verbindung über Gewichte zu jedem Neuron der Kohonen-Karte (Abbildung 4.9). Jedes Neuron wird durch einen Gewichtsvektor  $\vec{w}$  repräsentiert. Eine Art der Visualisierung ist die Komponentendarstellung (*Component Planes*), welche diese Eingangsknoten-spezifischen Gewichte in einer Flächendarstellung abbilden. Diese und weitere Darstellungsformen (z.B. *U-Matrix*) der Gewichte sind in [FLB08] beschrieben. Die Auswertung der Neuronen-Cluster erfolgt durch eine Zustands-/ Klassendarstellung (*Hit-Map*). Hierbei werden durch einen Kennzeichnungs-Prozess (*labeling*) die Cluster den Musterarten zugeteilt und die für jeden Cluster aktiven Neuronen gekennzeichnet. Dies kann zum einen selbstständig - durch einen Clusteralgorithmus - erfolgen und zum anderen aber auch durch ein überwachtes Verfahren, bei dem muster- bzw. klassenspezifische Eingabewerte der SOM präsentiert und die erregten Cluster markiert werden.

### Aufbau und Algorithmus

Die Self-Organizing-Map wird durch  $n$  Neuronen  $w_1, \dots, w_n$  aufgebaut. Diese werden jedoch durch ihren Gewichtsvektor  $\vec{w}_1, \dots, \vec{w}_n$  repräsentiert. Die Initialisierung der Gewichte kann entweder zufällig oder durch bestehende Erfahrungswerte erfolgen. Jeder Gewichtsvektor  $\vec{w}$  besteht aus  $x$ -Elementen. Diese sind äquivalent zu der Anzahl  $m$  der Eingabemerkmale  $x_1, \dots, x_m$ . Die Anordnung der Neuronen kann in unterschiedlicher Weise erfolgen, z.B. als Linie, zweidimensionale Karte oder dreidimensionale Karte. Die Gewichte der Neuronen werden durch eine Lernregel mit Schrittweitenfaktor an den Eingabe-Merkmalraum angepasst ( $f : \vec{w}_k \rightarrow \vec{x}$ ). Das zu adaptierende Neuron wird anhand folgender Lernregel an den Merkmalraum angepasst:

$$\vec{w}_k(n+1) = \vec{w}_k(n) + \alpha \cdot (\vec{x}(n) - \vec{w}_k(n)) \quad (4.33)$$

Die Gleichung 4.33 zeigt die Anpassung des Gewichtsvektors  $\vec{w}_k$  zum Zeitpunkt  $n$ , wobei  $k$  das ausgewählte Neuron (Gewinnerneuron) repräsentiert.  $\alpha$  ist die Schrittweite bzw. Lernrate, welche im Bereich  $0 < \alpha < 1$  in Abhängigkeit des Anwendungsfalls (es gibt keine allgemeingültige Vorschrift

zur Bestimmung von  $\alpha$ ) gewählt wird. Der Eingabevektor  $\vec{x}(n) \in \mathbb{R}^m$  ist ein zufällig ausgewählter Trainingsvektor aus der Datensatzmenge (*batch learning*) mit  $m$  Merkmalen. Durch die Anpassung der Gewichte - den Lernvorgang - wird das ausgewählte Neuron in die Richtung des Eingabevektors verschoben (siehe Abbildung 4.10).

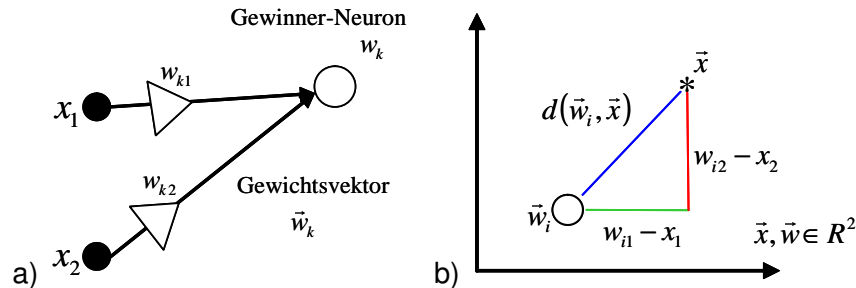


Abbildung 4.10: a) Die Merkmale  $x_1$  und  $x_2$  des zufällig aus dem Datensatz ausgewählten Musters werden der SOM über die Eingabeknoten präsentiert. b) Distanzmaß: Euklidische Distanz - entspricht dem Satz des Pythagoras bei  $m = 2$

Der Lernvorgang wird nach Erreichen einer gewünschten Epochenzahl beendet. Für die Auswahl des aktuell zu trainierenden Neurons wird ein Distanzmaß (z.B. Euklidische Distanz, siehe Abschnitt 4.4.2 bzw. Gleichung 4.34) verwendet. In Abbildung 4.10b ist die Berechnung der Euklidischen Distanz geometrisch dargestellt. Es fällt auf, dass diese bei  $m = 2$  dem Satz des Pythagoras entspricht, wobei die Summe der quadrierten Katheten (grün und rot) gleich der quadrierten Hypotenuse ist. In dem SOM-Algorithmus wird durch Berechnung der geringsten Distanz zu  $\vec{x}$  das Neuron ausgewählt, welches den minimalen Abstand zum Trainingsvektor aufweist.

$$d(\vec{w}_i, \vec{x}) = \sqrt{\sum_{j=1}^m (w_{ij} - x_j)^2} \quad (4.34)$$

Das Neuron  $w_k$ , welches - anhand des Distanzmaßes nachgewiesen - die geringste Distanz zu  $\vec{x}$  aufweist wird als Gewinnerneuron bezeichnet. Wenn jedoch nur ein Neuron adaptiert, so entstehen zwei zu lösende Probleme:

- Die abgebildete Neuronen-Karte ist nicht Topologie-erhaltend.
- Der Eingaberaum wird nicht zusammenhängend durch die Neuronen abgedeckt.

In Abbildung 4.11 ist dieses Problem anhand einem Beispiel dargestellt. Es fällt auf, dass nur partiell Neuronen adaptiert werden. 4 Neuronen sollen die zwei Cluster C1 und C2 abbilden. Es wird jeweils nur das Gewinnerneuron  $w_k$  mit dem geringsten Abstand zu dem zufällig ausgewählten Vektor  $\vec{x}$  adaptiert. Nach 4 Epochen werden die Cluster jedoch nur durch 2 Neuronen abgebildet. Neuron 1 und 3 werden nicht adaptiert.

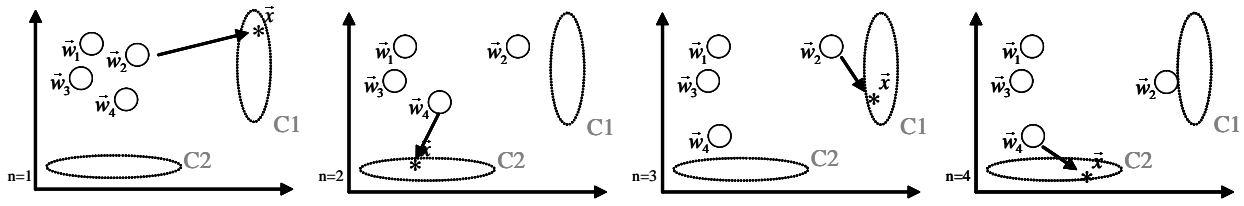


Abbildung 4.11: Problematik des Adaptionsvorgangs nach Gleichung 4.33

Aus diesen Gründen sind die Neuronen der Karte durch eine Nachbarschaftsfunktion  $\varphi \in \mathbb{Q}$  (*Neighborhood function* [Koh01]) miteinander verknüpft. Durch diese Funktion wird ein Nachbarschaftsradius um das ausgewählte (dem Eingangsvektor nächsten) Neuron definiert. Dadurch wird nicht nur das ausgewählte Neuron mit dem minimalsten Abstand, sondern auch die Neuronen in der Nachbarschaft auf den aktuellen Eingabevektor angelernet. Die direkt angrenzenden Neuronen haben den Abstand  $R = 1$ , die im nächst größeren Radius befindlichen Neuronen  $R = 2$  und so weiter. Für die Berechnung des Abstands  $R$  zwischen den Neuronen wird ebenfalls eine Distanzfunktion  $d(w_i, w_k)$  genutzt. Diese ist symmetrisch, so dass  $d(w_i, w_k) = d(w_k, w_i)$  gilt.

Der Grad, zu dem die angrenzenden Neuronen durch die Lernregel in Richtung des Trainingsvektors verschoben werden, hängt von der Nachbarschaftsfunktion  $\varphi(w_i, w_k) \leq R$  im Wertebereich  $[0, 1]$  ab. Die Bewegung der Neuronen  $w_i$  mit einer großen Distanz wird dadurch geringer ausfallen als die Bewegung der Neuronen mit einer geringen Distanz zu dem aktiven Neuron  $w_k$ .

In den meisten Anwendungen wird folgende Treppen-Nachbarschaftsfunktion benutzt:

$$\varphi(w_i, w_k) = \begin{cases} \left( \frac{1}{d(w_i, w_k)} \right); & d(w_i, w_k) \leq R \\ 0; & \text{sonst} \end{cases} \quad (4.35)$$

Eine rechenintensivere, jedoch feiner abgestufte Variante der Nachbarschaftsfunktion, stellt die in Gleichung 4.36 gegebene Berechnung über die Gaußsche-Glockenfunktion dar.

$$\varphi(w_i, w_k) = \begin{cases} e^{-\left( \frac{d(w_i, w_k)^2}{2\sigma^2} \right)}; & d(w_i, w_k) \leq R \\ 0; & \text{sonst} \end{cases} \quad (4.36)$$

Für beide Gleichungen wird der Abstand vom Gewinnerneuron  $w_k$  zu den anderen Neuronen  $w_i$  durch folgende Gleichung bestimmt:

$$d(w_k, w_i)^2 = \sum_{j=1}^n (w_k - w_{ij})^2 \quad (4.37)$$

Dabei stellt  $n$  die Dimension der Neuronenkarte ( $\mathbb{R}^n$ ) dar. In Abbildung 4.12 sind zwei verschiedene Topologien abgebildet und jeweils mit den Abstandsmaßen nach Gleichung 4.37 versehen (in a, b als Zahl in den Neuronen, in c als dritte Dimension). In [Sei01, Sei98, SM97, SM95] werden beispielsweise mehrdimensionale SOM's zur Bildverarbeitung genutzt.

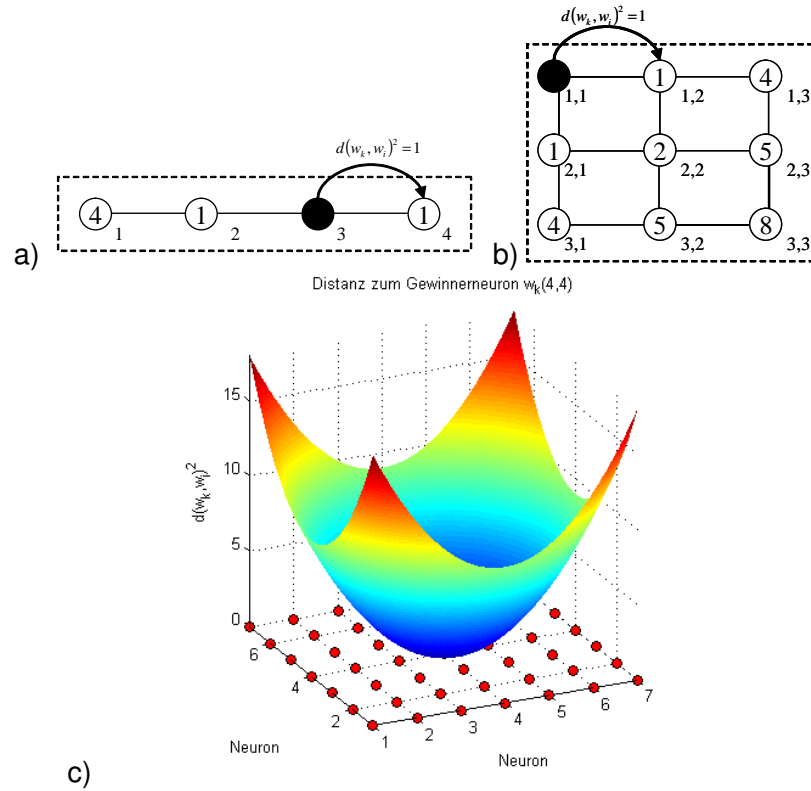


Abbildung 4.12: Beispiele zur Berechnung der Distanz vom Gewinnerneuron (schwarz) zu den Nachbarneuronen:

- a) Eine Linien-Topologie ( $\mathbb{R}^1$ )
- b) Eine Netz-Topologie ( $\mathbb{R}^2$ )
- c) Eine Netz-Topologie, jedoch mit  $7 \times 7$  Neuronen (rote Punkte, Gewinnerneuron ist  $w_k(4,4)$ )

Durch Integration der Nachbarschaftsfunktion in die Lernregel der Gleichung 4.33 werden die Neuronengewichte des Gewinner-, als auch der Nachbarneuronen durch folgende Formel aktualisiert:

$$\vec{w}_i(n+1) = \vec{w}_i(n) + \alpha \cdot \varphi(w_i, w_k) \cdot (\vec{x}(n) - \vec{w}_i(n)) \quad (4.38)$$

Um eine gleichmäßige Verteilung der Neuronen im Raum zu erreichen, wird ein zeitlich veränderlicher Blockradius  $r(t)$  benötigt. Dabei gibt es zum Beispiel folgende Realisierungsvarianten:

- $r(t)$  als lineare/ nichtlineare mit der Zeit fallende Funktion.
- $r(t)$  als Teil der Nachbarschaftsfunktion:

$$\varphi(w_i, w_k, t) = e^{-\left(\frac{d(w_i, w_k)^2}{2\sigma(t)^2}\right)} \quad (4.39)$$

mit

$$\sigma(t) = \sigma_S + \frac{t}{T} (\sigma_E - \sigma_S) \quad (4.40)$$

Die zeitabhängige Scheitelbreite (Abbildung 4.13) wird von einem Startwert  $\sigma_S$  ausgehend über  $T$  Trainingsdurchläufe auf den Endwert  $\sigma_E$  reduziert. Eine mit der Zeit abnehmende Scheitelbreite der Nachbarschaftsfunktion führt zu einer geringeren Adaption der Nachbarneuronen. Dargestellt sind die Nachbarschaftsfunktionen für  $\sigma = 1$  und  $\sigma = 0.2$  bezogen auf das Gewinnerneuron  $w_k(4,4)$  für ein  $7 \times 7$  Neuronen-Netz.

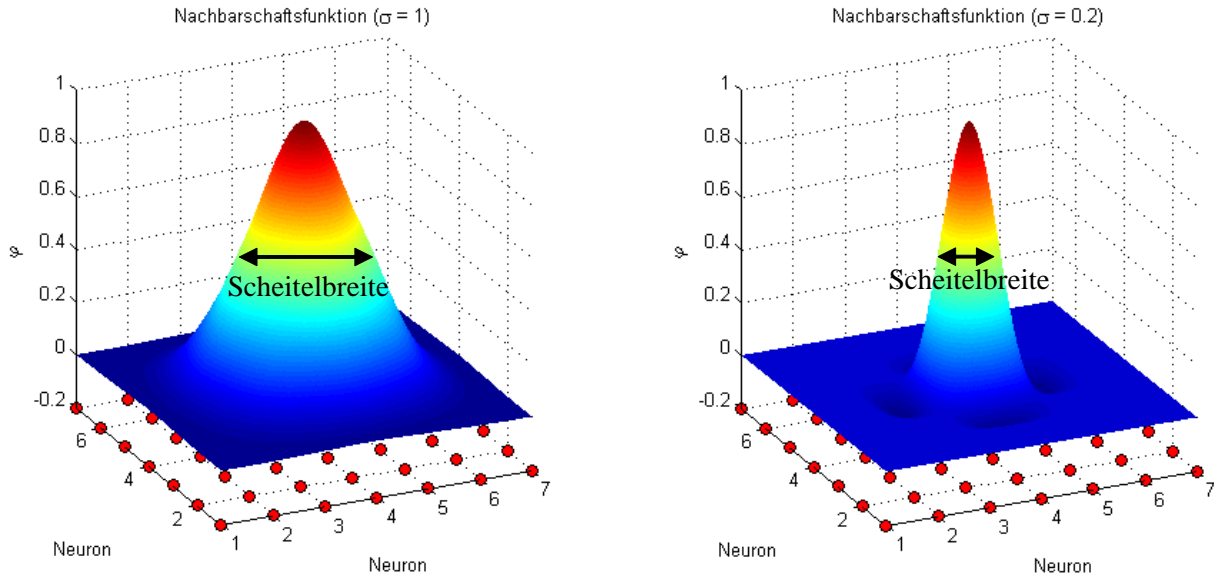


Abbildung 4.13: Einfluss der Scheitelbreite der Nachbarfunktion

Eine weitere Verbesserung der Anpassung der Neuronengewichte erfolgt durch die Betrachtung des aktuellen Zeitpunkts bzw. Integration eines Zeitparameters. In den meisten Fällen wird die SOM über eine definierte Anzahl von Trainingsschritten/ Epochen an den Eingaberaum angepasst. Eine Optimierung erfolgt durch eine schnelle Anpassung der Neuronen am Anfang und einer präzisen Annäherung an das Optimum (die Häufungspunkte des Eingaberaums) zum Ende des Trainings. Dies wird durch eine mit der Zeit  $t$  abnehmende Schrittweite  $\alpha(t)$  erreicht.

$$\alpha(t) = \alpha \left( 1 - \frac{t}{T_E} \right) \quad (4.41)$$

Die Änderung der Schrittweite erfolgt für Gleichung 4.41 linear, kann jedoch auch nichtlinear implementiert werden.  $T_E$  stellt die Trainingsdauer bzw. Epochenzahl dar. Eine weitere lineare Umsetzung mit Definition eines Start- und Endwertes wäre äquivalent zu Gleichung 4.40. Die finale Gleichung zur Adaption der Gewichte zum Zeitpunkt  $n$  ergibt sich somit wie folgt:

$$\vec{w}_i(n+1) = \vec{w}_i(n) + \alpha(n) \cdot \varphi(w_i, w_k, n) \cdot (\vec{x}(n) - \vec{w}_i(n)) \quad (4.42)$$

### Darstellung und Auswertung

Für die Auswertung des adaptierten Kohonen-Netzes werden verschiedene Auswerteformen benutzt. Eine Zusammenfassung über die gebräuchlichsten gibt es in [FLB08]. Prinzipiell gibt es zwei wichtige Faktoren, die Aufschluss über den Zustand der SOM geben:

- die Ausprägung der Gewichte für jedes Neuron in Abhängigkeit vom Eingabevektor (Komponentendarstellung),
- die Klassenzuordnung der Neuronen mit maximaler Aussteuerung bei definierter Anregung (Klassendarstellung).



Durch einen Test-/ Markierungs-Datensatz werden die reagierenden Neuronen ermittelt und gekennzeichnet. Die Kennzeichnung kann zum Beispiel durch Farbe (Abbildung 4.14) oder ein Tortendiagramm erfolgen. Das Tortendiagramm bildet die Auftrittshäufigkeiten der einzelnen Klassen für jedes Neuron ab (*Hit-Map*).

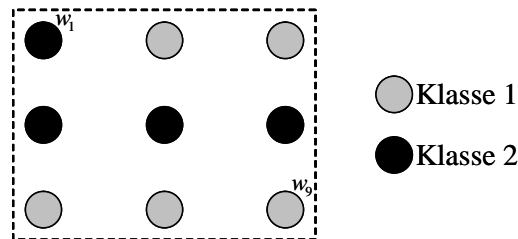


Abbildung 4.14: Die Zuordnung der Klasseninformation zu den Neuronen erfolgt durch eine Zustands-/ Klassendarstellung.

In der Komponentendarstellung (*Component Layer/ Map/ Plane*) werden die Gewichtsvektoren eingangswertbasierend aufgeteilt und in zwei Dimensionen abgebildet. Für jeden Eingabewert  $x$  wird dadurch eine zweidimensionale Ebene gebildet, welche sich in Abhängigkeit der verwendeten Neuronen aufspannt. Die Amplitude des Gewichtes an jedem Koordinatenpunkt wird über eine Farbskala abgebildet. Durch den Kennzeichnungsprozess kann den Neuronen der Komponentendarstellung auch eine Klasseninformation zugeordnet werden. Diese Form der Auswertung wird genutzt um den Eingaberaum der Neuronen auszuwerten und dadurch z.B. auch abzuleiten, wie groß der Einfluss eines Eingangsmerkmals des Eingabevektors  $\vec{x}$  auf jedes Neuron ist.

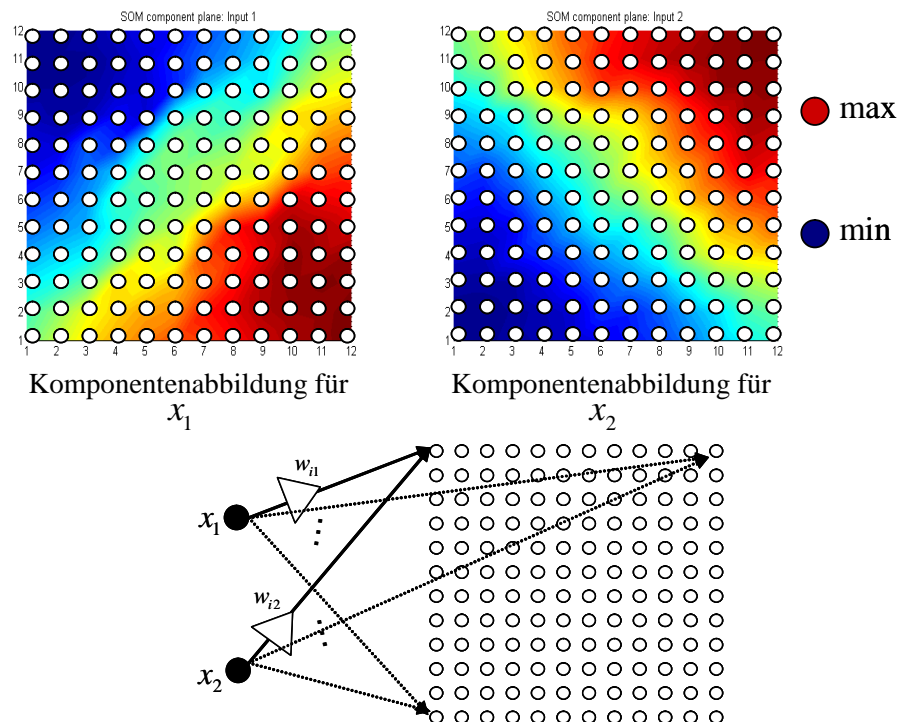


Abbildung 4.15: Schematische Darstellung eines 12x12 Neuronennetz durch eine Komponentendarstellung

In Abbildung 4.15 (wo die Gewichte der Neuronen für  $x_1$  über die Farbinformation dargestellt wer-

den) ist ersichtlich, dass das Neuron (1,12) eine deutlich stärkere Ausprägung hat als Neuron (12,1), bezogen auf eine Anregung über das erste Eingangsmerkmal des Eingabevektors  $\vec{x}$ . Durch einen quantitativen Vergleich der  $m$  Komponentenschichten können der Einfluss jedes Eingabemerkmals  $x_1, \dots, x_m$  auf die Neuronen analysiert werden.

Dementsprechend teilt sich die Interpretation der Komponentendarstellung für die in Abbildung 4.9 gezeigten SOM-Typen wie folgt auf:

- 2D-Netzstruktur: Jedes Merkmal des Eingabevektors  $\vec{x}$  bildet eine Komponentenschicht der Dimension  $[n \times n]$ .
- 3D-Netzstruktur: Jedes Merkmal des Eingabevektors  $\vec{x}$  bildet  $n$  Komponentenschichten der Dimension  $[n \times n]$ .

Alternativ können auch über Cluster-Algorithmen (z.B. k-Means oder diverse Abstandsmaße, siehe Abschnitt 4.4.2) die Neuronenanhäufungen im Eingaberaum identifiziert und markiert werden. Dadurch ist theoretisch eine unüberwachte Einteilung/ Zuordnung unbekannter Daten möglich. In den meisten Fällen werden jedoch nur dann gute Ergebnisse erreicht, wenn die Zentroide der Neuronensammlung der zu erkennenden Zustände sich nur wenig überlappen, also durch getrennte Punktwolken im Raum definiert sind.

Die genannten Darstellungsformen ermöglichen die einfache geometrische Analyse von hochdimensionalen, komplexen, nichtlinearen Daten. Basierend auf der Kohonen-Karte sind eine Vielzahl von aufbauenden Algorithmen entwickelt worden, wie z.B. die Neural Gas SOM oder die Fuzzy SOM. Als Beispiel für eine komplexe Anwendung wird in [Kan94] die Anwendung der SOM für Spracherkennungsaufgaben und Sprachqualitätsbewertung diskutiert. Eine Veranschaulichung verschiedener selbstorganisierender Algorithmen durch ein Java-Demo ist in [DG] gezeigt.

#### Fehlerfunktion

Der SOM-Algorithmus beinhaltet keine Fehlerfunktion. Die Konvergenz der Neuronenkarte hängt in erster Linie von der Anzahl der Trainingsschritte und den Parametern der Nachbarschaftsfunktion ab. Die Adaption nach Gleichung 4.42 erfolgt in Abhängigkeit der definierten Anzahl von Trainingsschritten, dennoch gibt es Möglichkeiten, die Güte der Neuronenausprägung im Eingaberaum zu bewerten. Eine Möglichkeit stellt der Quantisierungsfehler dar (siehe auch [Neu98]). Dabei erfolgt die Bewertung der Adaptionsgüte durch einen mittleren Fehler, welcher den Abstand des Gewichtsvektors des feuерnden Neurons (Gewinnerneuron)  $\vec{w}_{kn}$  zu dem aktuell gewählten Merkmalvektor  $\vec{x}_n$  eines Testdatensatzes mit  $N$  Mustern berechnet.

$$e = \frac{1}{N} \sum_{n=1}^N \min [d(\vec{x}_n, \vec{w}_{kn})] \quad (4.43)$$

Der Quantisierungsfehler  $e$  ist groß, wenn die durch einen Testdatensatz aktivierten Neuronen im Mittel weit entfernt von den Anregungspunkten sind. Er ist klein, wenn die Neuronen einen kleinen Abstand zu den Anregungspunkten aufweisen. Der Fehler dient jedoch nicht als absolutes Maß, wie gut die Neuronenkarte konvergiert ist, sondern vielmehr als relativ zu betrachtendes Maß, ob

die aktuell berechnete Karte besser adaptiert ist als zuvor. Der Quantisierungsfehler kann also beispielsweise zur Bestimmung der notwendigen Trainingsepochen verwendet werden.

### Time-Delay Self-Organizing Map

Diese Art der selbstorganisierenden Karten verwendet Methoden, die eine zeitliche Veränderung der Eingangsparameter in der Auslegung des *Self-Organizing Maps* Algorithmus berücksichtigen. Neben den Beziehungen zwischen den Merkmalen, die durch den Werteraum der Neuronengewichtsvektoren  $\vec{w}_k$  gebildet werden, wird auch die Beziehung zu der Zeit bei der Adaption der Neuronengewichte berücksichtigt. Dies kann durch folgende Methoden erreicht werden:

- Verzögerungsglieder im Eingabebereich der Kohonenkarte (*Delay-Line*)
- *Leaky Integration*, ein Faktor zur zeitlichen Beeinflussung der Gewichtungsfaktoren der Kohonenkarten-Neuronen
- Rekurrente Kohonenkarte

In [Kan94] sind die Ansätze für den *Self-Organizing Map* Aufbau von *Time-Delay*- als auch *Leaky-Integration*-Verfahren mit Experimenten zur Spracherkennung aufgezeigt. In [Ves97] findet die *Time-Delay-SOM* ebenfalls Anwendung in der Prädiktion von Zeitreihen. Eine weitere interessante Anwendung zur Phonem-Erkennung ist in [WHH<sup>+</sup>89] gegeben. Auch hier wird eine Verzögerungsstufe am Eingang des Neuronalen Netzes für die Auswertung des zeitlichen Signalverlaufs auf die Ausbildung der Neuronengewichte und der damit verbundenen Erkennungsrate (gegenüber dem Hidden-Markov-Modell HMM) genutzt.

Die Analyse zeitlich veränderlicher Daten findet neben der Spracherkennung auch in der Erkennung von Objekten Anwendung. Durch die Integration eines *Leaky Integrator Layers* in [SB98] wird das Neuronale Netz mit einem Verlaufsspeicher versehen. Jedes Neuron weiß demzufolge, wann bzw. wie oft es aktiviert wurde, wodurch sich der zeitlich veränderliche Pfad in dem durch die Neuronenvektoren abgebildeten Werteraum nachvollziehen lässt. Die ursprüngliche Idee wurde erstmals in [RT91, CT93] vorgestellt. Bei der Mustererkennung von Phonemen wurde in [Kan94] bei Verwendung der Integratoren eine Verringerung der Fehlerrate um ca. 7% erreicht. Das durch die Integratoren erzeugte Übertragungsverhalten entspricht einem Tiefpassfilter, welches am Eingang und/ oder Ausgang jedes Neurons implementiert wird.

Bei rekurrenten Kohonennetzen ist der Aufbau ähnlich zu dem MLP-Netz (*Multi-Layer Perceptron*) mit *error-backpropagation*-Algorithmus. Ein zusätzlicher Gewichtsvektor speichert die Information der Neuronenkarte aus den vorangegangenen Lernschritten und ermöglicht somit eine Rückkopplung auf den aktuellen Adaptionprozess der Neuronengewichte. Aus allen Quellen geht jedoch hervor, dass es kein universell einsetzbares Kohonenkartenmodell für alle Mustererkennungsprobleme gibt. Jeder einzelne Fall benötigt in der Regel ein gesondert abgestimmtes Klassifizierungsmodell ([HN06]).

In dem bisher vorgestellten SOM-Modell wurden durch die gewählten Merkmale und die Struktur der Neuronenkarte der Werteraum durch die Neuronen abgebildet. Wie aus [Voe02] bekannt, kann zusätzlich der Zeitbereich mit erfasst werden. Dazu wird das SOM-Modell beispielsweise um eine

integrierte Verzögerungsstufe erweitert. In Abbildung 4.16 ist die integrierte Verzögerungsstufe im Eingabebereich der Kohonenkarte gezeigt.

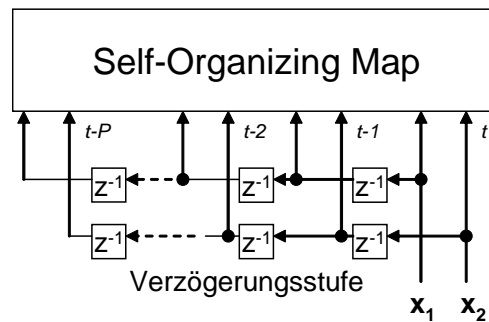


Abbildung 4.16: Aufbau der Verzögerungsstufe;  $P$  Verzögerungsglieder für jedes Merkmal des Eingabevektors  $\vec{x}$

Durch die Verzögerungen werden also bei Betrachtung der Komponentenschichten die Veränderungen der Merkmalausprägung im Werteraum zwischen den Merkmalen zum Zeitpunkt  $t = 0$  bis  $t = P$  betrachtet. Die Konstante  $P$  definiert die Anzahl der verwendeten Verzögerungsstufen. Ein Eingabevektor mit beispielsweise 2 Merkmalen wird dementsprechend wie folgt gebildet:

$$\vec{x} = \{x_1(t), x_2(t), x_1(t-1), x_2(t-2), \dots, x_1(t-P), x_2(t-P)\} \quad (4.44)$$

#### 4.4.5 Transparente recheneffiziente Klassifikationsalgorithmen

Wie bereits am Anfang dieses Kapitels erwähnt, existiert eine Vielzahl an Klassifikationsverfahren. Neuronale Netze und insbesondere selbstorganisierende Karten wurden bereits vorgestellt. Die Verfahren unterscheiden sich neben ihrer Struktur und Arbeitsweise auch in dem benötigten Rechenaufwand und ihrer Transparenz. Dies bedeutet in diesem Zusammenhang, dass ein Algorithmus mit guter Transparenz die Entscheidungswege leicht interpretierbar zur Verfügung stellt. In [Ruf09] sind der Bayes- und Fuzzy-Klassifikator in ihrem Rechenaufwand und der Nachvollziehbarkeit der Entscheidungswege bewertet. In diesem Abschnitt sind weitere Verfahren genannt und die Ergebnisse aus [Ruf09] zusammengefasst dargestellt.

- **Lernende Vektorquantisierung**

Wie der SOM-Algorithmus stammt auch der LVQ-Algorithmus (*Learning Vector Quantization*, LVQ) von Kohonen und stellt ein überwachtes Lernverfahren dar. Als Vorreiter der SOM weist dieser auch eine ähnliche Struktur auf. Eine Menge an zufällig initialisierten Gewichtsvektoren werden anhand einer Lernregel an einen Trainingsmerkmalraum angepasst. Ein neuer oder bereits trainierter Eingabevektor wird parallel mit allen Gewichtsvektoren verglichen, und derjenige, der ihm am nächsten ist, legt seine Klasse fest.

- **Entscheidungsbäume**

Diese können auch als regelbasierte Klassifikatoren bezeichnet werden. Sie sind eine spezielle Darstellungsform von Entscheidungsregeln und veranschaulichen die aufeinanderfolgenden hierarchischen Entscheidungskriterien. Ein Entscheidungsbaum (*Decision Tree*) mit

seinen Entscheidungswegen wird beispielsweise durch den CART-Algorithmus (*Classification and Regression Trees*) von [BFOS84] berechnet.

- **k-Nächste-Nachbarn-Klassifikator**

Dieser Algorithmus ist ein Klassifikationsverfahren, bei dem unter Berücksichtigung des Abstandes zu den  $k$  nächsten Nachbarn (*k-nearest-neighbor*, k-NN) eine Klassenzuordnung durchgeführt wird. Die Nachbarn werden durch Gewichtsvektoren im Merkmalraum repräsentiert, welche im Lernvorgang definiert wurden. Die Berechnung kann durch diverse Distanz-Metriken erfolgen (siehe Abschnitt 4.4.2). Der Faktor  $k$  bewirkt eine Mehrheitsentscheidung. Die Klassenentscheidung fällt demzufolge in Abhängigkeit der mehrheitlich vorliegenden  $k$  Nachbarn der gleichen Klasse.

- **Verfahren der größten Wahrscheinlichkeit (*Maximum-Likelihood*)**

Dieses Verfahren wird vorwiegend als *Maximum-Likelihood*-Klassifikator (ML-Klassifikator) bezeichnet, basierend auf Wahrscheinlichkeit. Dabei wird - im Lernvorgang - für alle Muster einer jeweiligen Klassen die Wahrscheinlichkeitsdichtefunktion (Verteilungsdichtefunktion) gebildet. Wie in Abbildung 4.17 gezeigt ist, wird ein unbekanntes Muster - im Testvorgang - durch Bildung von Entscheidungsräumen einer Klasse zugeordnet. Die Bildung von zwei Entscheidungsräumen durch den ML-Klassifikator erfolgt durch die gepunktete Linie. Der ML-Klassifikator stellt ein einfaches Verfahren zur Klassifikation dar, welches vor allem in Anwendungen eingesetzt wird, wo eine Fehlklassifikation für alle Klassen die gleiche Konsequenz hat.

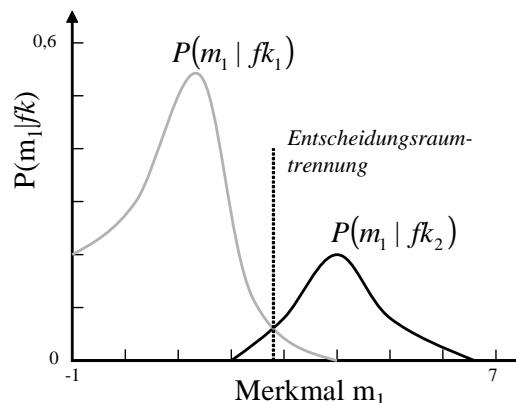


Abbildung 4.17: Eindimensionale Verteilungsdichtefunktion des Merkmals  $m_1$  für zwei Fehlerklassen  $fk_1$  und  $fk_2$

- **Support-Vector-Machine**

Hierbei handelt es sich um ein Klassifikationsverfahren (SVM), welches die Trennebene (Hyperebene) zur Trennung der Muster in den Klassen so optimiert, dass möglichst große Freiräume (ohne Objekte) um diesen Klassengrenzen entstehen. Die Trainingsmuster, welche sich in nächster Nähe zu der Trennebene befinden, werden als Stützvektoren (woran die Namensgebung erfolgt ist) bezeichnet, da nur diese zur Berechnung einer optimalen Trennebene verwendet werden. Jedoch können nur linear trennbare Datensätze durch eine Hyperebene

getrennt werden. Es existieren dennoch *Kernel*-basierte *Support-Vector-Machines* (siehe [Bur98, SS01]), welche durch Transformation des Merkmalraums in einen höherdimensionalen Raum auch nichtlineare Datensätze trennen.

##### • Fuzzy-Klassifikatoren

Basierend auf unscharfen (*fuzzy*) Mengen entscheiden diese Algorithmen. In [Ruf09] wurde dieser Klassifikator als recheneffektiver Klassifikator, mit der Prämisse einer möglichst guten Transparenz, untersucht. In seiner Arbeit werden die Prozesse zur Entscheidungsgenerierung:

- Fuzzifizieren (Eingangsgrößen erhalten Zugehörigkeitsfunktionen basierend auf beispielsweise Maximalwert, Minimalwert, Mittelwert oder Standardabweichung),
- Regelverknüpfung,
- und Defuzzifizieren

detailliert beschrieben. Gerade die Zugehörigkeitsfunktionen können mit unterschiedlichem Rechenaufwand implementiert werden. So können bereits einfache Dreiecks-Zugehörigkeitsfunktionen gute Klassifizierungsergebnisse bei sehr geringem Rechenaufwand erreichen; eine Gauß'sche Zugehörigkeitsfunktion (nach Carl Friedrich Gauß) ist hingegen mit einem höheren Rechenaufwand verbunden.

##### • Bayes-Klassifikator

Dieser Klassifikatortyp, benannt nach Thomas Bayes, ist ein auf Wahrscheinlichkeit bzw. dem Bayes-Theorem<sup>4</sup> basierender Mustererkenner. Dafür muss theoretisch für jede Klasse die Wahrscheinlichkeitsdichtefunktion vorhanden sein. Je nach Anzahl der Trainingsmuster in den Klassen kann diese jedoch auch durch eine Normalverteilung beschrieben werden, wobei anhand weniger Anlern Daten deren Parameter geschätzt werden. Eine besondere Realisierungsvariante stellt der naive Bayes-Klassifikator dar, welcher aufgrund seines geringen Rechenaufwands gern verwendet wird, solange die Merkmale des Datensatzes wenig korreliert sind. Das Prinzip des naiven Bayes-Klassifikators (NB-Klassifikator) ist mit dem des ML-Klassifikators ähnlich. Sie unterscheiden sich jedoch in ihrer Entscheidungsfunktion. Der ML-Klassifikator entscheidet anhand der bedingten Wahrscheinlichkeit  $P(m | fk)$  - wobei  $m$  das Merkmal und  $fk$  die Fehlerklasse definieren (siehe Abbildung 4.17) - und der NB-Klassifikator anhand der Entscheidungsregel  $P(fk | m)$ . Für den naiven Bayes-Mustererkenner wird also die Wahrscheinlichkeit berechnet, mit der eine Fehlerklasse  $fk$  den Merkmalvektor  $m$  enthält. In [Ruf09] wurde dieser Algorithmus aus den genannten Gründen mit dem Fuzzy-Klassifikator in Bezug auf Mustererkennung - mit minimalem Rechenaufwand und hoher Transparenz - in akustischen Datensätzen angewendet.

In Tabelle 4.1 sind auszugsweise Ergebnisse aus der Arbeit von [Ruf09] dargestellt. Dabei wurde ein Datensatz, basierend auf Mikrofonaufnahmen, mit 8 Klassen definiert. Die Klassen unterscheiden sich in der gefahrenen Geschwindigkeit und den Straßenbelägen. Die Klassifikationsaufgabe besteht darin, die Klassen zu trennen und dadurch eine Entscheidung des aktuell vorliegenden

---

<sup>4</sup>Definiert das Rechnen mit bedingten Wahrscheinlichkeiten, basierend auf der bedingten Wahrscheinlichkeit, dass ein Ereignis A unter der Bedingung B eintritt.

Fahrbahnbelags zu ermöglichen. Weiterführend soll dabei jedoch ein möglichst effizientes Klassifikationsverfahren eingesetzt werden, welches einen geringen Rechenaufwand und eine hohe Transparenz (Nachvollziehbarkeit des Entscheidungswegs) aufweist. Dafür wurden zwei potentielle Verfahren ausgewählt, der Fuzzy- und der naive Bayes-Klassifikator. Es ist in der Tabelle ersichtlich, dass mit diesen beiden Mustererkennern sehr gute Ergebnisse erreicht werden. Eventuell ist die gute Trennbarkeit bereits bei wenigen Merkmalen auf eine zu geringe Datenmenge und damit unzureichender statistischer Sicherheit zurückzuführen. Es wurden 1036 Muster verwendet, jedes entspricht einem Zeitbereich von 250 ms. Als Vergleich wurde ein Neuronales Netz herangezogen, welches ebenfalls sehr gute Ergebnisse erreicht. Die drei Verfahren unterscheiden sich schlussfolgernd kaum in Ihren Erkennungsraten. Es ist jedoch deutlich ersichtlich, dass sich die effizienten Klassifikatoren (Fuzzy und Bayes) deutlich in der Rechenzeit vom neuronalen Netz unterscheiden. Der Bayes-Klassifikator ist ungefähr 7 mal schneller, der Fuzzy-Klassifikator sogar 14 mal schneller als das KNN bei Verwendung von 3 Merkmalen. Dieser Vorteil reduziert sich jedoch bei Verwendung von mehr Merkmalen, dennoch bleibt dieser bestehen. Es wurde also gezeigt, dass sich sehr gute Ergebnisse mit wenig Rechenaufwand und guter Nachvollziehbarkeit der Entscheidungswege erreichen lassen. Dennoch geht auch hervor, dass die Klassifikatoren Datensatzabhängig sind und die hier gezeigten Ergebnisse nicht allgemeingültig auf alle Datensätze übertragen werden können. Weiterführende Informationen und Vergleiche mit weiteren Datensätzen sind in [Ruf09] gegeben.

	"FUZZY 3"	"BAYES 2"	"NEURONALES NETZ"
<b>3 MERKMALE</b>			
⊙ <i>Erkennungsrate in %</i>	100.00	96.41	94.83
<i>Rechenzeit in s</i>	7.51	15.78	105.75
<b>9 MERKMALE</b>			
⊙ <i>Erkennungsrate in %</i>	100.00	98.13	99.94
<i>Rechenzeit in s</i>	14.31	33.54	74.22

Tabelle 4.1: Vergleich der Erkennungsraten und Rechenzeiten, aus [Ruf09]

In Abschnitt 5.3.1 sind ebenfalls verschiedene Klassifikatortypen für die Datensätze, die in dieser Arbeit zur Entwicklung eines Agenten zur Fehlerindikation generiert wurden, bewertet; jedoch unter dem Gesichtspunkt einer möglichst hohen Erkennungsrate und guten Allgemeingültigkeit (Datensatzunabhängigkeit).





## 5 Berechnung des akustischen Fingerabdrucks

Grundlage dieser Arbeit stellt das Sammeln von akustischen Daten dar. In vielen Evolutionsstufen wurde eine Datenbank aufgebaut. Erste Datenbanken für Grundlagenversuche wurden in Matlab® erstellt. Aufgrund der enormen Datenmengen (zur Erzeugung von akustischen Fingerabdrücken muss eine gewisse statistische Sicherheit in den Messdaten erreicht werden.), bestehend aus Rohdaten (10 Sekunden bei 48 kHz Abtastrate und 16-Bit digitalisiert benötigen ca. 1 Megabyte Speicherplatz.), vorverarbeiteten Daten (aus den digitalisierten Mikrofonsignal werden durch Transformationen Merkmale berechnet) und beschreibenden Daten (Protokolldaten), wurde im nächsten Schritt eine auf SQL basierende PostgreSQL-Datenbank erzeugt (ein freies Open Source Datenbanksystem, siehe Anhang A.1, Abbildung A.1). Die Datenbankstruktur ist universell benutzbar, so dass jede Messung unabhängig ihres eigentlichen Gebrauchs in der Datenbank abgelegt werden kann. Die Kommunikation mit der Datenbank erfolgt durch verschiedene Schnittstellen. Die Daten können mit Matlab® über die JDBC-Schnittstelle (*Java Database Connectivity Advanced Protokoll Interface*) abgerufen, ausgewertet und verarbeitet werden. Des weiteren steht die Schnittstelle Npgsql (*Open Source .Net Data Provider for Postgresql*) zur Verfügung, so dass mit der Programmiersprache C# (*C-Sharp*) effiziente Datenbank-verknüpfte Programme entwickelt werden können. In Kapitel 7 werden die gesammelten Erkenntnisse dieser Arbeit zur Erzeugung von Daten, deren Auswertung und automatischen Bewertung (siehe Kapitel 6) erfasst und ein mobiles akustisches Diagnosesystem in C# entworfen. Dieses soll die Aufnahme, Analyse und Ablage möglichst einfach gestalten und gleichzeitig Schnittstellen zu Matlab® aufweisen, so dass beispielsweise neue Algorithmen einfach und portabel getestet werden können. In diesem Kapitel wird die Messdatenvorverarbeitung und -auswertung behandelt, mit dem Ziel, einen akustischen Fingerabdruck (eindeutig beschreibende Merkmale für die Fehlergeräuschklassen) zu erzeugen.

### 5.1 Beschreibung der Datensätze

Die Fehlerdatenbank der Konzernforschung enthält tausende Fahrzeug- und Akustikdaten. Im Rahmen der folgenden Auswertungen wurden Messdaten aus einem VW Polo 1.4l ausgewertet. Diese Daten liefern Informationen über 12 verschiedene Fehlerzustände, siehe Tabelle 5.1.

Klangbild/ Fehlerzustand	Klasse	Anzahl Signalausschnitte (gut hörbar)
defektes Motorlager	1	65
Koppelstange ausgeschlagen rechts	2	60
Koppelstange ausgeschlagen links	3	85
Radlager stark beschädigt rechts	4	40
Radlager stark beschädigt links	5	40
defektes Radlager links	6	25
defektes Radlager rechts	7	19
Koppelstange lose geschraubt links	8	50
defekte Gelenkwelle Außenlager rechts	9	48
defektes Federbeinlager rechts	10	25
defektes Federbeinlager links	11	17
Fehlerfrei/ Originalklang	12	170

Tabelle 5.1: Übersicht Fahrzeugmessdaten VW Polo 1.4l

Es ergeben sich vorerst für die Auswertung in Summe 644 Signalausschnitte, welche sich durch eine gute Hörbarkeit des angegebenen Fehlerzustands definieren. Die Hörbarkeit wurde subjektiv ermittelt, nähere Erläuterungen zum Vorgehen sind in [Wor08, Kot08] zusammengefasst. Die Datensätze bestehen aus fahrzeugbeschreibenden CAN-Bus-Daten (*Controller Area Network*) wie z.B. Lenkwinkel oder Giergeschwindigkeit, und akustischen Messdaten. Aus den akustischen Messdaten wurden zustandsbeschreibende Merkmale (Features) berechnet. Diese lassen sich untergliedern in:

- physikalische Merkmale (RMS/ Effektivwert)
- spektrale Merkmale (Leistungsspektrum, Oktavspektrum, Diskrete-Kosinustransformation, etc.)
- psychoakustische Merkmale (Lautheit, Rauigkeit, Artikulationsindex, etc.)

Die Berechnung und Extraktion der Merkmale, welche sich durch ein- und mehrwertige unterscheiden lassen, erfolgte für jeden der 644 Signalausschnitte, wobei diese meist eine Signallänge von ca. 1.5 bis 2.5 Sekunden aufweisen. Eine Übersicht über die resultierenden Merkmale und deren Komplexität ist in Tabelle 5.2 aufgezeigt. Dabei stellt  $N$  die Anzahl der Fenster dar, welche variabel, in Abhängigkeit von der Datensatzlänge sind.

Im nächsten Schritt erfolgt die Vorauswahl aller berechneter Merkmale für die Generierung verschiedener Testdatensätze. Die 644 hörbaren Abschnitte haben keine fixe Länge. Deshalb wurde der Parameter  $t(n)$ , die Signalausschnittlänge, eingeführt. Diese stellt eine Zeitkonstante dar, welche die gut hörbaren Sequenzen in Abschnitte gleicher Länge unterteilt. Dies ist nötig, um eine konstante Mittelungszeit für die Berechnung der Merkmale zu schaffen. Die konstante Signallänge  $t(n)$  wurde variiert, und die mehrwertigen Merkmale wurden über diese Zeit gemittelt. Für ein großes  $t(n)$  enthält der erzeugte Testdatensatz demzufolge stark gemittelte Werte. Für ein kleines  $t(n)$  verhält es sich entsprechend umgekehrt. Ziel ist es dabei, die Merkmalselektion und Klassifikation mit verschiedenen Merkmalsätzen aus den erzeugten Testdatensätzen zu testen und die Variation der Ergebnisse auch in Abhängigkeit der Merkmalsatzart und -komplexität zu analysieren. Die Komplexität ergibt sich aus der Berechnungsmethode des jeweiligen Merkmals. Die Berechnung

Psychoakustische Merkmale	Spektrale/ Physikalische Merkmale	CAN Merkmale
Spez. Schwankungsstärke [1x21]	Spektrum [2047xN]	Lenkgeschwindigkeit [1xN]
Spez. Schwankungsstärke vs. Zeit [21xN]	Gemitteltes Spektrum [2047]	Lenkwinkel [1xN]
Artikulationsindex vs. Zeit [1xN]	Cepstrum [4096xN]	Geschwindigkeit [1xN]
Schwankungsstärke vs. Zeit [1xN]	Terzspektrum [31xN]	Giergeschwindigkeit [1xN]
Modulationsfrequenz vs. Zeit [1xN]	Oktavspektrum [10xN]	
Kurtosis vs. Time [1xN]	MFCC [13xN]	
Lautheit vs. Zeit [1xN]	DCT [500xN]	
Rauhigkeit vs. Zeit [1xN]	Diskrete Wavelet-Koeffizienten [4x4096xN]	
Schärfe vs. Zeit [1xN]	Crestfaktor [1xN]	
Tonhaltigkeit vs. Zeit [1xN]	RMS [1xN]	
Spezifische Lautheit [1x240]	Spektrale Schwankungsstärke [1xN]	
Spezifische Lautheit vs. Zeit [240xN]	Spektraler Centroid [1xN]	
Spezifische Rauhigkeit [1x24]	Spektraler Rolloff [1xN]	
Spezifische Rauhigkeit vs. Zeit [24xN]		

Tabelle 5.2: Übersicht berechneter Merkmale [Dimension]

des Effektivwerts (RMS - *Root Mean Square*) zum Beispiel benötigt im Vergleich zur Berechnung des Spektrums nur einen Bruchteil der Rechenzeit. Weiterhin lassen sich diese durch ihre Art in physikalische, spektrale und CAN-Bus-Merkmale unterteilen. Für die Untersuchungen wurden vier verschiedene Datensätze generiert:

- Datensatz *ALL* (VW\_MW\_ALL\_L):
  - Dimension des Merkmalsatzes: [9327+4]
  - Anzahl der Merkmalsätze:  $[N(t(n))]$
  - Enthaltene Merkmale:
    - \* Spalte 1,2 und 3 der in Tabelle 5.2 gegebenen Merkmale (außer Wavelet-Koeffizienten)
- Datensatz *PSY* (VW\_MW\_PSY\_L)
  - Dimension des Merkmalsatzes: [578+4]
  - Anzahl der Merkmalsätze:  $[N(t(n))]$
  - Enthaltene Merkmale:
    - \* Spalte 1 und 3 der in Tabelle 5.2 gegebenen Merkmale
- Datensatz *SEL* (VW\_MW\_SEL\_L)
  - Dimension des Merkmalsatzes: [13+4]
  - Anzahl der Merkmalsätze:  $[N(t(n))]$
  - Enthaltene Merkmale:
    - \* Alle Einwertigen Merkmale ([1xN]) der in Tabelle 5.2 gegebenen Merkmale
- Datensatz *SPEC* (VW\_MW\_SPEC\_L)
  - Dimension des Merkmalsatzes: [8749]
  - Anzahl der Merkmalsätze:  $[N(t(n))]$
  - Enthaltene Merkmale:
    - \* Alle Merkmale aus Spalte 2 der in Tabelle 5.2 gegebenen Merkmale (außer Wavelet-Koeffizienten)
- weitere gemischte Datensätze *UNI*, *ALL-SPEC*

Die Datensatzlänge, welche sich aus der Anzahl der Merkmalsätze ergibt, wird nach folgendem Verhältnis (mit  $t(n)$  als definierte Signalauschnittlänge, z.B. 200 Millisekunden) berechnet:

$$N(t(n)) = \frac{\text{Sequenzlaenge}_{\text{GutHoerbar}}(n)}{t(n)}$$

Die Merkmale wurden vorerst nur für ein Mikrofon berechnet, welches sich auf der Fahrerposition befand. Die Datensätze spiegeln somit den Höreindruck des Fahrers wider. Prinzipiell unterscheiden sich die Merkmale, wenn man die aufgenommenen Mikrofondaten vom Fahrer- und

Beifahrersitzplatz vergleicht. Dies ist neben der Innenraumakustik des Fahrzeugs natürlich auch auf die unterschiedlichen Schallemissionsquellen zurückzuführen. Um den Einfluss der Signalauschnittlänge auf die gemittelten Merkmale zu untersuchen, wurden für die oben genannten vier Merkmalsätze verschiedene Variationen in der Zeit getestet, siehe Tabelle 5.3.

Parameter	Zeitkonstante in ms			
$t(n)$	250	500	1000	var
$N(t(n))$	4760	2380	1190	644
Deklaration:	$T025$	$T05$	$T1$	$Tvar$

Tabelle 5.3: Variation der Signalauschnittlänge  $t(n)$  und resultierende Datensatzlänge  $N(t(n))$

Betrachtet man die Häufigkeitsverteilung der Klassen in dem Datensatz mit variabler Signalauschnittlänge, zeigt sich ein klares Ungleichgewicht (siehe Anhang A.1, Abbildung A.2 und Abbildung A.3).

Theoretisch gesehen würde dies bedeuten, dass der Klassifikator in der Trainingsphase mit zehnmal höherer Wahrscheinlichkeit auf einen Merkmalsatz mit der zugehörigen Fehlerklasse 12 angelernt wird gegenüber der Fehlerklasse 11. Demnach erfolgt die Adaption der Gewichte vielmehr auf die Fehlerklasse 12 als auf Fehlerklasse 11. Um dies zu verhindern, werden die Datensätze so manipuliert, dass eine Gleichverteilung, also jede Fehlerklasse mit der gleichen Wahrscheinlichkeit, auftritt.

## 5.2 Normierung der Merkmale

Als Voraussetzung für eine erfolgreiche Klassifikation müssen alle Merkmale normalisiert werden; damit wird eine ungewollte Gewichtung bei der Abbildung der Werte auf den Merkmalraum und dessen Abstandsmaße verhindert. Die in Gleichung 5.1 gegebene *z-score* Normalisierung berechnet sich aus dem aktuellen Eingabewert  $x(n)$ , dem Mittelwert  $\mu$  und der Standardabweichung  $\sigma$ .

$$s(n) = \frac{x(n) - \mu}{\sigma} \quad (5.1)$$

$$\mu = \bar{x} = \frac{1}{N} \sum_{n=1}^N x(n) \quad (5.2)$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (x(n) - \bar{x})^2} \quad (5.3)$$

Ein weiteres Verfahren ist die Min-Max Normalisierung nach Gleichung 5.4. Wobei  $max_{new}$  und  $min_{new}$  die wählbaren Grenzen des neuen Wertebereichs sind.

$$s(n) = \frac{x(n) - min_x}{max_x - min_x} (max_{new} - min_{new}) + min_{new} \quad (5.4)$$

Die z-score wird auch als Standardnormalverteilung oder z-Transformation bezeichnet. Alle Normalverteilungen können durch die beschriebene Transformation auf einen Standard mit  $\mu = 0$  und  $\sigma = 1$  gebracht werden. Man spricht deshalb von der Standardnormalverteilung (siehe [Pap08, Lar05]).

### 5.3 Anwendung der Selektionsverfahren

Die wichtigsten im Kapitel 4.3 gezeigten Verfahren zur Merkmalreduktion werden auf die im Abschnitt 5.1 gegebenen Datensätze angewendet. Die Vorarbeiten von [Kot08] hinsichtlich der Anwendbarkeit der *Fisher-Diskriminante* haben deutlich gezeigt, dass dieses Verfahren sehr gut für die Reduktion der Merkmale in einem zweistelligen Bereich geeignet ist. Für eine weitere Verringerung des Rechenaufwands werden folgende Merkmalreduktionsverfahren, aufbauend auf der *Fisher-Diskriminante*, miteinander verglichen:

- *Fisher-Diskriminante*
- Korrelationskoeffizient
- *Relief* Algorithmus
- Transinformation (*Mutual Information*)
- Informationsgewinn (*Information Gain*)
- *Fast-Correlation-Based-Filter* (FCBF)

#### 5.3.1 Bewertung der Selektionsverfahren

Die zu untersuchenden Merkmalreduktions- bzw. Selektionsverfahren sind alle den so genannten *Filter-Methoden* zugeordnet, siehe Kapitel 4.3. Die Grundlage dieser Methodik ist die Reduktion der Dimensionen des Datensatzes, jedoch ohne in direkter Verbindung zu einem Klassifikator zu stehen. Alle Filter-Methoden berechnen ein Maß zur Bewertung der einzelnen Merkmale des Datensatzes. Aufgrund dieser Bewertung kann eine Gewichtung und dadurch auch eine Auswahl der besten Merkmale erfolgen. Trotz der Unabhängigkeit der Verfahren zu den Klassifikatoren kann eine Auswahl für ein Reduktionsverfahren nur mit Hilfe eines Klassifikators getroffen werden. Der Klassifikator berechnet dabei die erreichbare Güte des reduzierten Datensatzes. Die Güte oder auch Klassifikationsrate (siehe Gleichung 3.1) steht dabei als Maß, wie gut der niedrigdimensionale Datensatz eine Trennung der Klassen zulässt.

Aus der Literatur sind zahlreiche Gegenüberstellungen von Klassifikatoren und *Filter-Methoden* bekannt, welche den Schluss zulassen, dass Klassifikatoren und Selektionsverfahren je nach betrachtetem Datensatz unterschiedliche Ergebnisse bringen, siehe [Kot08, PLD05]. Aus verschiedensten Literaturquellen, wie zum Beispiel [Guo05, DP05], gehen dabei die Support-Vector-Machines (SVM) als auch die Neuronalen-Netze (NN), vorwiegend das Multi-Layer-Perceptron-Netz, als zuverlässige und allgemein anwendbare überwachte Mustererkennungsverfahren hervor. Zur Validierung der aus der Literaturrecherche hervorgehenden hauptsächlich verwendeten Klassifikatoren wurde

eine Auswahl von sechs verschiedenen Mustererkennungsverfahren miteinander verglichen, siehe folgende Auflistung:

- *Classification Tree* (Tree, [MM01])
- *K-Nearest-Neighbor Classifier* (k-NN, [Hu01])
- *Learning-Vector-Quantization* (LVQ, [HH06])
- *Maximum-Likelihood-Classifer* (ML, [Hu01])
- *Support-Vector-Machine* (SVM, [Caw00])
- *Multilayer-Perceptron-Neural-Network* (MLP oder MLP-NN, [Hu01])

Die Algorithmen wurden ausgehend von den genannten Quellen in Matlab® an das Datensatzformat von *UCI* [AN07] angepasst. Weiterhin wurde in dem MLP-Algorithmus eine Validierungsprozedur implementiert, welche eine Überanpassung (*overfitting*) des neuronalen Netzes verhindert. Alle verwendeten Datensätze wurden mit der *z-score*-Normierung im Wertebereich angepasst.

Zur Berechnung der durchschnittlichen Güte der Klassifikatoren wurden fünf verschiedene Datensätze von der *UCI Machine Learning Repository* ([AN07]) verwendet, siehe Tabelle 5.4.

Datensatz	Merkmale	Klassen	Datensatzlänge
Iris	4	3	150
Wine	13	3	178
Yeast	8	10	1484
TAE	5	3	151
Statlog (Vehicle)	18	4	846

Tabelle 5.4: Übersicht über die Testdatensätze der *UCI Machine Learning Repository*

Die Auswahl der Datensätze erfolgte mit dem Anliegen, eine möglichst große Variation in der Anzahl der Klassen, Merkmale und der Datensatzlänge zu erreichen. Interessant dabei ist auch der unterbestimmte Fall des *Yeast* Datensatzes, wo wenige Merkmale zur Erkennung von vielen Klassen verwendet werden. Dieser Fall wird auch für die Anwendung eines akustischen Mustererkenners im KFZ vorliegen. Aufgrund des möglichst minimal zu haltenden Rechenaufwands müssen nur wenige Merkmale ausreichen, um diverse Fehlergeräusche zu erkennen. Bei der Anwendung der Klassifikatoren, siehe Abbildung 5.1, haben sich die Ergebnisse aus der Literatur bestätigt. Die SVM- und MLP-NN-Mustererkenner erreichen die beste Performance, siehe Tabelle 5.5.

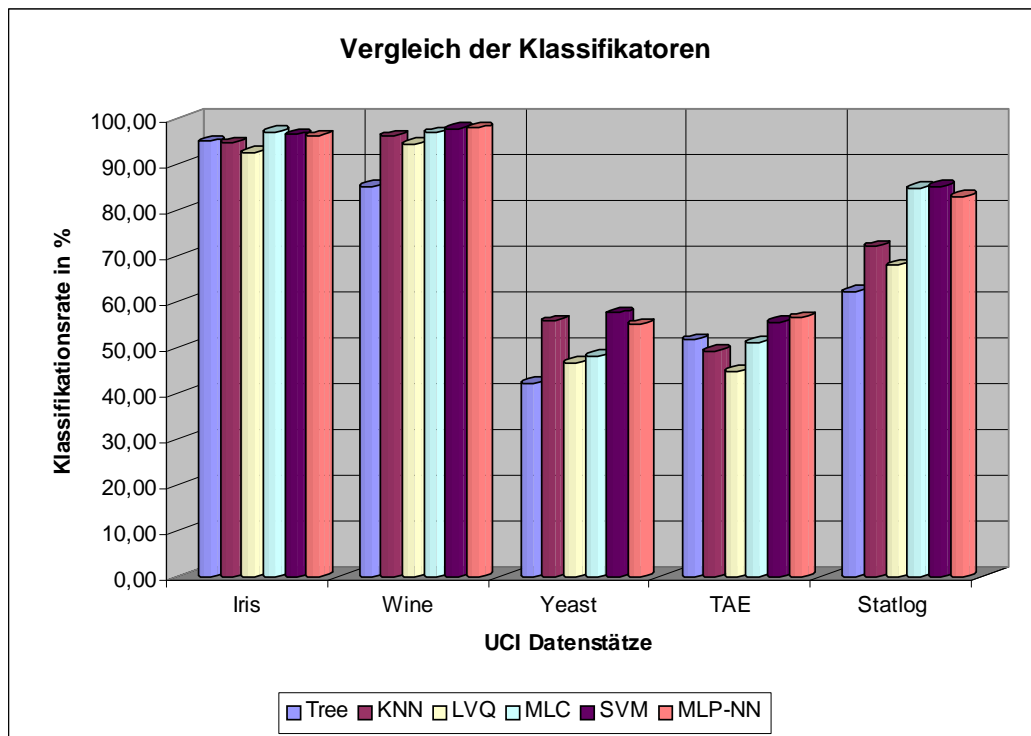


Abbildung 5.1: Vergleich verschiedener Mustererkennungsverfahren auf der Basis von UCI Datensätzen

Die Mustererkenner wurden auf 70% der Testdatensatzlänge angelernt. Die Auswahl der Trainingsmenge erfolgte nach dem *Bootstrapping*-Verfahren, siehe Unterabschnitt 4.4.1. Die Testdatenmenge von 30% diente dann zur Berechnung der Klassifikationsrate, welche das Verhältnis von richtig ermittelten Klassen zu den falsch ermittelten Klassen wiedergibt. Um ein repräsentatives Ergebnis zu bekommen wurde die Klassifikation, bestehend aus Datenaufteilung, Trainings- und Testphase, für jeden Datensatz jeweils 25 mal durchgeführt.

	Tree:	k-NN:	LVQ:	ML:	SVM:	MLP:
Iris	94.89	94.62	92.62	97.07	96.36	96.09
Wine	85.17	96.14	94.34	96.83	97.74	97.89
Yeast	42.27	55.83	46.69	48.16	57.53	55.08
TAE	51.73	49.34	44.89	51.11	55.56	56.53
Statlog	62.12	72.10	67.87	84.71	85.09	82.98
Mittelwert	67	74	69	76	78	78

Tabelle 5.5: Klassifikationsraten *acc* in % (gemittelt über 25 Durchläufe) für die UCI Datensätze und die mittlere erreichte Klassifikationsrate für das jeweilige Mustererkennungsverfahren über alle Datensätze

Um die Performance der Klassifikatoren hinsichtlich der erreichten Güte zu bewerten, betrachtet man die mittlere Klassifikationsrate der Mustererkenner über alle Datensätze. So gibt es zwischen dem SVM und MLP Algorithmus kaum Unterschiede. Beide erreichen eine mittlere Güte von rund 78 Prozent, gefolgt vom ML mit 76 Prozent. Betrachtet man die Güte der einzelnen Datensätze, dann können diese zwei besten Verfahren jedoch eine um bis zu über 10% höhere Klassifikationsrate erreichen, welches am deutlichsten bei dem *Yeast*- und *Statlog*-Datensatz wird.



Für die Bewertung der Selektionsverfahren wurde deshalb der MLP-NN Klassifikator ausgewählt. Dieser wird auf einer Trainingsmenge des verwendeten Datensatzes angelernet. Für die Berechnung der Klassifikationsgüte wird eine Testmenge verwendet. Das Verhältnis zwischen Trainings- und Testdatenmenge wird durch den *ratio*-Faktor festgelegt. Nach jedem Trainingsdurchgang wird der Backpropagation-Algorithmus mit einem zufällig ausgewählten Merkmalsatz auf die momentan erreichte Klassifikationsrate getestet. Wenn diese stagniert oder ansteigt (*overfitting*), wird der Trainingsprozess beendet. Anschließend wird die Klassifikationsrate des trainierten neuronalen Netzes mit der Testdatenmenge ermittelt. Für eine repräsentative Klassifikationsgüte wird über mehrere Durchgänge gemittelt (*folds*).

#### Parameter des neuronalen Netzes:

- Das Training des MLP-NN wurde jeweils mit einer Trainingsdatenmenge von 70% des zu untersuchenden Datensatzes durchgeführt (*ratio*=70%).
- Die Berechnung der Klassifikationsrate wurde über 5 Durchgänge gemittelt (*folds*=5).
- Es wurden 2 verborgene Neuronenschichten implementiert.
- Die verborgenen Neuronenschichten enthalten 40% mehr Neuronen gegenüber der Anzahl der Neuronen in der Eingabeschicht, welche sich aus der Merkmalanzahl ergibt.

### 5.3.2 Fisher-Diskriminante

In diesem Abschnitt wird die *Fisher*-Diskriminante (im folgenden FD), siehe Abschnitt 4.3.2, auf die 4 Datensätze aus Abschnitt 5.1 angewendet. Aus den Untersuchungen von [Kot08] geht hervor, dass sich die FD hervorragend zur Reduktion der Merkmaldimension eignet. Unter Berücksichtigung der fahrzeugspezifischen Daten (CAN-Merkmale) ließen sich für fast alle Klassenkombinationen Merkmale finden, welche eine FD-Wert  $f > 2$  erreichen und somit eine gute Trennung der Klassen darstellen. Weiterhin wurde eine sinnvolle Bewertung bzw. Gruppierung der FD-Werte vorgenommen, siehe Tabelle 5.6.

FD-Wert	Trennbarkeit
$f \leq 1$	schlecht
$1 < f \leq 2$	mittel
$f > 2$	gut

Tabelle 5.6: Bewertung der *Fisher*-Diskriminanten-Trennbarkeit durch Einteilung in drei Gruppen

Eine Beispielauswertung [Kot08] der FD für die CAN-Merkmale (siehe Tabelle 5.2) ist in Tabelle 5.7 gegeben. In dieser Tabelle sind oberhalb der Symmetrieachse die FD-Werte eingetragen und unterhalb die Indizes des zugehörigen CAN-Merkmals (Indizes äquivalent zur Reihenfolge in Tabelle 5.2). Die FD-Werte zeigen deutlich auf, dass durch die CAN-Merkmale die Klassen Federbeinlager links und rechts gut von den anderen Klassen trennbar sind. Aus den dazugehörigen CAN-Merkmalen geht hervor, dass ein CAN-Merkmal logischerweise für verschiedene Klassenkombinationen den größten FD-Wert liefert. Theoretisch ergibt sich die Anzahl der Merkmale in dem FD-Merkmalsubset aus Gleichung 5.5, wobei  $K$  die Anzahl der Fehlerklassen darstellt.

$$M_{FDmax} = \frac{(K^2 - K)}{2} \quad (5.5)$$

$M_{FDmax}$  entspricht somit dem durch die FD-Berechnung reduzierten Merkmalsatzumfang. Sollte ein Merkmal für mehrere Klassenkombinationen jeweils den höchsten *Fisher*-Wert darstellen, so reduziert sich die Anzahl der wichtigen Merkmale weiter. Die FD ist aufgrund der einfachen Berechnungsvorschrift und des damit verbundenen geringen Rechenaufwands sehr gut zur Reduktion der Merkmalmenge geeignet. Eine Bewertung der Merkmale hinsichtlich ihres Einflusses auf die Klassifikationsgüte (Ermittlung einer Rangfolge der Merkmale, *ranking*) ist jedoch nicht möglich.

	MotorL	KopR	KopL	RadRs	RadLs	RadL	RadR	KopLos	Gelw	FedbL	FedbR	Original
MotorL		0,32	0,15	0,83	0,39	3,8	0,46	0,06	9,67	17,53	24,71	0,22
KopR	3		0,05	1,31	0,32	6,17	1,31	0,17	64,61	17,51	53,14	0,36
KopL	3	2		1,3	0,32	6,16	0,78	0,15	64,12	16,92	52,49	0,36
RadRs	2	4	4		0,32	0,55	0,2	1,32	0,74	12,73	1,12	0,16
RadLs	3	4	4	2		1,56	0,55	0,32	2,37	17,33	6,06	0,09
RadL	4	4	4	4	4		0,35	6,2	0,39	22,67	1,89	1,07
RadR	1	1	1	3	1	4		0,63	0,89	19,38	2,33	0,19
KopLos	2	3	1	4	4	4	1		65,01	17,88	52,96	0,36
Gelw	4	4	4	4	4	3	3	4		66,77	0,35	1,42
FedbL	1	1	1	1	1	3	3	1	4		7	8,54
FedbR	2	2	2	4	2	2	3	2	4	4		3,09
Original	2	4	4	2	1	4	1	4	4	1	2	

Tabelle 5.7: Trennbarkeit durch FD bei ausschließlicher Verwendung der CAN-Merkmale (Mittelwert, variable Zeitkonstante, aus [Kot08])

Die Ergebnisse der *Fisher*-Diskriminanten-Auswertung sind in Tabelle 5.8 zusammengefasst. Die verwendeten Datensätze unterscheiden sich durch:

- die Zeitkonstante  $t(n)$  (linker Bereich),
- die Merkmalauswahl (mittlerer Bereich, siehe Abschnitt 5.1).

Im rechten Bereich der Tabelle 5.8 ist die jeweilige Anzahl der FD-Merkmale  $M_{FD}$  für die vier Datensätze gegeben. Ein Vergleich der Zeitkonstanten mit der Klassifikationsgüte der einzelnen Datensätze zeigt, dass die Güte steigt, wenn die Zeitkonstante abnimmt. Eine abnehmende Zeitkonstante bedeutet, dass die Merkmalberechnung für einen kleinen Zeitabschnitt erfolgt. Bei der Mittelung der mehrwertigen Merkmale geht somit die Tendenz zum aktuellen *Ist*-Wert. Gleichzeitig stehen dem Klassifikator mit abnehmender Zeitkonstante mehr Merkmalsätze zum Training, Validierung und Test zur Verfügung. Es ist somit anzunehmen, dass eine kleinere Zeitkonstante  $t(n)$  für die Berechnung der Merkmale eine genauere akustische Abbildung der Fehlerklassen zulässt und somit die Erkennungsrate erhöht.

Zeitkonstante $t(n)$	Datensatzgüte						Anzahl der FD Merkmale $M_{FD}$					
	ALL	PSY	SPEC	SEL	UNI	ALL-SPEC	ALL	PSY	SPEC	SEL	UNI	ALL-SPEC
variabel	91.48	90.16	91.15	70.82	90.82	84.92	41	22	40	13	27	26
1000 ms	86.89	84.26	86.89	69.51	90.82	88.85	43	24	44	14	24	25
500 ms	95.41	94.59	95.90	79.02	95.57	92.95	43	29	44	12	30	28
250 ms	99.02	97.55	98.61	79.84	98.69	96.57	41	27	40	13	28	25

Tabelle 5.8: Güte des MLP-NN Mustererkenners für ausgewählte Fisher-Diskriminante-Merkmale (FD)

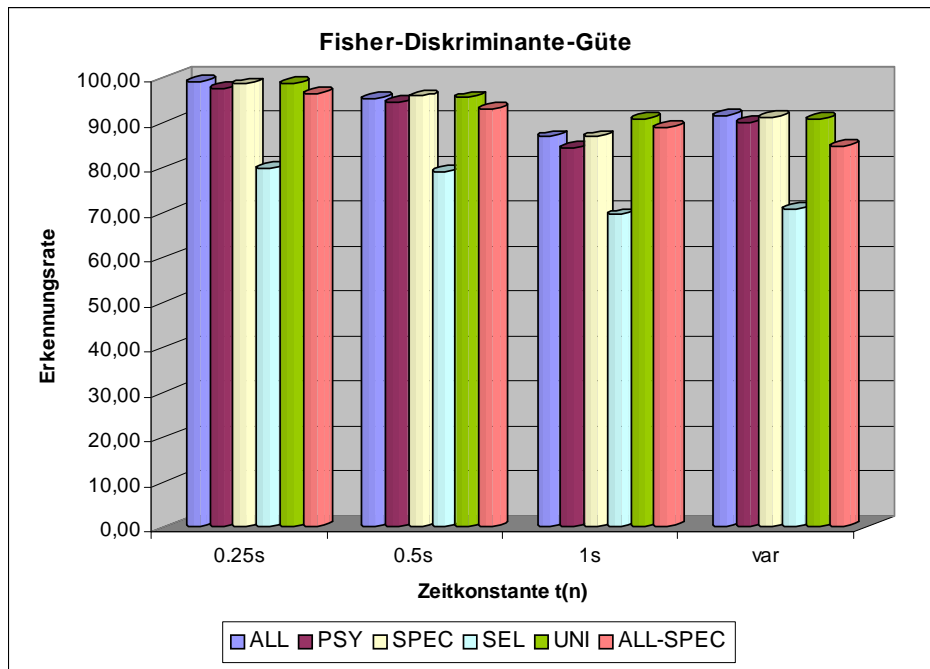


Abbildung 5.2: Klassifikationsraten bei Merkmalreduktion durch die *Fisher*-Diskriminante bei unterschiedlicher Zeitkonstante  $t(n)$

Der Datensatz *ALL*, welcher aus insgesamt 9331 Merkmalen besteht, erreicht im Vergleich zu den anderen drei Datensätzen die höchste Güte. Betrachtet man diese für die variable Zeitkonstante, so können 8% gewonnen werden, wenn die Zeitkonstante auf 250 ms verringert wird. Betrachtet man die FD-Matrix für diesen Fall, so lässt sich erkennen, dass fast alle Fehlerklassenkombinationen eine  $f$ -wert größer 2 erreichen. Einzig die Kombinationen zwischen Klasse 1 und 2 (Motorlager, Koppelstange ausgeschlagen rechts) gegenüber Klasse 3 (Koppelstange ausgeschlagen links) erreichen einen Wert  $f$  kleiner 2. Die abgebildete FD-Matrix ergibt sich insgesamt aus 39 verschiedenen Merkmalen.

Bei der Klassenmatrix für den *PSY* Datensatz treten vereinzelt mittelmäßig und schlecht trennbare Klassenkombinationen auf. Die abgebildete FD-Matrix ergibt sich insgesamt aus 26 verschiedenen Merkmalen.

Für den *SPEC* Datensatz ergibt sich ein fast analoges Abbild wie für den *ALL* Datensatz, selten treten mittelmäßig trennbare Klassenkombinationen auf; die Anzahl der Merkmale in der FD-Matrix ist ebenfalls 39. Vor allem die Fehlerklasse 1 und 2 lassen sich schlecht von Klasse 3 trennen.

Für den Datensatz *SEL* (Einzelmerkmale, siehe Abschnitt 5.1) gibt es einige schlecht und mittelmäßig trennbare Klassenkombinationen. Insgesamt wurden 14 Merkmale ermittelt, welche den höchsten FD-Wert für die jeweiligen Fehlerklassenkombinationen liefern. Aus Tabelle A.1 in Anhang A.1, welche die Wahrheitsmatrix darstellt, ist eine unregelmäßige Verteilung der falsch zugeordneten Fehlerklassen ersichtlich.

Vergleicht man die Klassifikationsrate der drei Datensätze *ALL*, *PSY* und *SPEC* bei  $t(n) = 250ms$  miteinander, so liegen diese gemeinsam im Bereich über 98%. Analog zu der optischen Auswertung über die Klassenmatrix äußert sich die schlechte Trennbarkeit in der Klassifikationsrate,

siehe Tabelle 5.8. Dieses Verhalten der schlechten Trennbarkeit ist dementsprechend auch in der Wahrheitsmatrix in Anhang A.1, Tabelle A.1 für den Datensatz *SEL* ersichtlich, wo Klasse 1 und 2 fälschlich als Klasse 3 und auch Klasse 5, 8 und 9 fälschlicherweise als Fehlerklasse 12 erkannt werden.

Zusammenfassend ergibt sich für den Datensatz *SEL*, welcher die Einzelmerkmale beinhaltet, nur eine mittelmäßige Klassifikationsrate im Vergleich zu den anderen Datensätzen. Für eine hinreichende Fehlerklassifikation müssen also auch andere Merkmale, wie z.B. mehrwertige psychoakustische, verwendet werden, um eine Güte größer 95% zu erreichen. Vor allem die Fehlerklassen 1, 2, 3, 5, 8 und 9 lassen sich durch die Einzelmerkmale nicht ausreichend genau bestimmen.

Eine geordnete Darstellung der extrahierten Merkmale für die FD-Berechnung enthält Anhang A.1, Tabelle A.2.

### 5.3.3 Fisher-Diskriminante und Korrelationskoeffizient

Der erste Ansatz für eine Reduktion der Merkmalvielfalt und Ordnung wurde mittels Korrelationskoeffizientenberechnung realisiert. Laut Abschnitt 4.3.4 definiert sich der Korrelationskoeffizient durch den Wertebereich  $r = [-1; 1]$ . Die folgenden Werte lassen sich wie folgt interpretieren:

- +1: perfekter linearer Zusammenhang,
- 0: gar kein linearer Zusammenhang,
- -1: perfekter gegensätzlicher linearer Zusammenhang.

Der Theorie zufolge sollten Merkmale, welche eine hohe Korrelation mit einem anderen Merkmal aufweisen, vernachlässigbar sein. Merkmale mit einer sehr kleinen Korrelation hingegen sind nicht „ersetzbar“:

- max. Korrelation signalisiert eine Ersetzbarkeit der Merkmale
- geringe bis mittlere Korrelation von Merkmal  $m_1$  zu  $m_2$  führt zu einer Rangfestlegung

Im ersten Schritt erfolgt die Berechnung des absoluten Korrelationskoeffizienten für jede mögliche Merkmalkombination. Der Absolutwert blendet die „Phaseninformation“ aus und stellt somit nur einen Wert für den linearen Zusammenhang zwischen zwei Merkmalen dar. Aus Abbildung 5.3 geht hervor, dass die entstehende Matrix laut Definition symmetrisch aufgebaut ist. Korrelationskoeffizienten für alle Merkmale des mit der *Fisher*-Diskriminante reduzierten Merkmalsatzes (*subset*). Dargestellt ist die Matrix für den Datensatz *PSY* bei  $t(n) = 250ms$ . Auf der Farbskala ist der Absolutwert des Korrelationskoeffizienten  $r$  abgebildet. Die Zuordnung der Merkmalindizes ist durch Tabelle 5.10 möglich. Auf der Diagonalen ist der Korrelationskoeffizient von einem Merkmal zu sich selbst aufgetragen. Dieser ist folglich +1.

Auf Basis dieser Matrix werden dann für jedes Merkmal die Kombination mit der höchsten Korrelation ermittelt, siehe Tabelle 5.9. Spalte 1 stellt den maximalen absoluten Korrelationskoeffizienten dar. In Spalte 2 und 3 sind die dazugehörigen Merkmale aufgelistet. Spalte 4 und 5 zeigen die Merkmalkombinationen auf, welche symmetrisch sind. Dies bedeutet beispielsweise, dass der maximale lineare Zusammenhang von Merkmal 3 mit dem Merkmal 2 besteht. Für Merkmal 2 gilt ebenfalls,

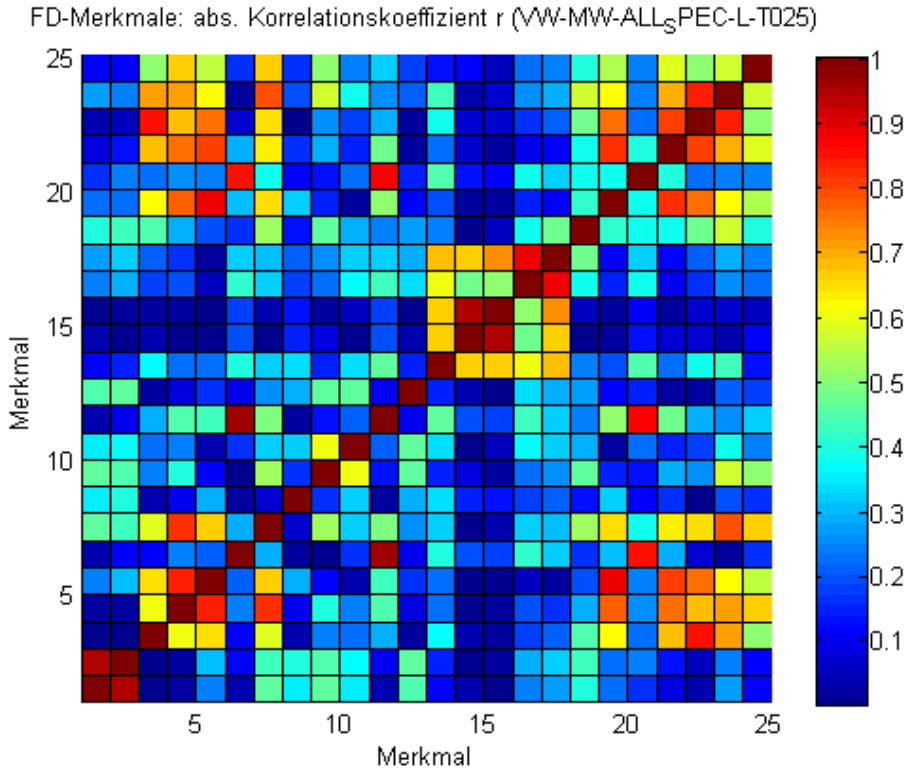


Abbildung 5.3: Absoluter Korrelationskoeffizient (Farbskala) für den reduzierten Merkmalsatz (*subset*)

dass der Korrelationskoeffizient nur mit Merkmal 3 maximal wird. Demzufolge sind Merkmal 2 und 3 bei dieser Betrachtung zueinander symmetrisch und folglich untereinander austauschbar.

Durch Einfügen eines Schwellenwerts wird die Grenze definiert, an der eine Merkmalkombination als „wichtig“ behandelt wird. Laut Theorie sollten die Merkmale einer Kombination mit einem Korrelationskoeffizienten gegen 0 in ihrer enthaltenen Information nicht korrelieren und demzufolge von hoher Priorität sein. Merkmale der Kombinationen mit einem Wert gegen 1 sind hingegen untereinander ersetzbar. In Abbildung A.5 im Anhang A.1, ist die reduzierte Kombinations-Matrix für  $r < 0.3$  dargestellt.

Für jedes Merkmal  $m$  kann nun ein mittlerer absoluter Korrelationskoeffizient  $|\bar{r}(m)|$  nach Gleichung 5.6 als Grundlage für eine Priorisierung bzw. Rangliste berechnet werden. Es wird der Mittelwert  $|\bar{r}(m)|$  über alle Merkmalkombinationen von Merkmal  $m$  zu den Merkmalen  $i$ , für die  $r < 0.3$  ist, berechnet.

$$|\bar{r}(m)| = \frac{1}{M} \sum_{i=1}^M |r(m, i)| \quad (5.6)$$

Der resultierende Vektor  $|\bar{r}(m)|$  wird anschließend sortiert, so dass das Merkmal mit dem kleinsten Wert an oberster Stelle steht und somit der höchsten Priorität entspricht. Das Merkmal mit der höchsten Priorität hat folglich den geringsten linearen Zusammenhang mit den anderen Merkmalen. Das Merkmal mit der geringsten Priorität hat einen großen Zusammenhang mit anderen Merkmalen.

$\max(r(m1,m2))$	m1	m2	m1_s	m2_s
0,9691	1	16	3	2
0,9937	2	3	5	4
0,9937	3	2	8	7
0,9196	4	5	11	9
0,9196	5	4	14	13
0,9433	6	7	16	15
0,9537	7	8	19	18
0,9537	8	7	22	21
0,9303	9	11		
0,8847	10	11		
0,9303	11	9		
0,5921	12	22		
0,9999	13	14		
0,9999	14	13		
0,9993	15	16		
0,9993	16	15		
0,9072	17	16		
0,9854	18	19		
0,9854	19	18		
0,8406	20	21		
0,8445	21	22		
0,8445	22	21		
0,6241	23	17		
0,4593	24	25		
0,6060	25	20		
0,5431	26	25		

Tabelle 5.9: Ermittlung des maximalen Koeffizienten von jedem Merkmal  $m1$  zu Merkmal  $m2$ 

Im nächsten Schritt lässt man die Erkenntnis über die austauschbaren Merkmale aus Tabelle 5.9 mit in die Priorisierung einfließen. Die Merkmale, welche auch durch ein anderes Merkmal abbildbar sind (symmetrische Merkmalkombinationen), sollten einen möglichst großen Abstand auf der Prioritätenliste haben. Dazu wird für jedes geordnete Merkmal der Reihe nach das symmetrische Merkmal gesucht und an das Ende verschoben. Für Merkmal 7 ist Merkmal 8 laut Tabelle 5.9 symmetrisch. Deswegen wird Merkmal 8 an das Ende verschoben und erhält somit eine niedrige Priorität in der Rangliste. Merkmal 13 hat einen starken linearen symmetrischen Zusammenhang mit Merkmal 14, deswegen wird letzteres an das Ende der Rangliste verschoben. Durch dieses Verfahren werden die Merkmale, welche durcheinander „austauschbar“ sind voneinander getrennt. Die resultierende Rangliste im Vergleich zu der ungeordneten *Fisher*-Diskriminanten Merkmalauswahl ist in Tabelle 5.10 dargestellt. Es erfolgt die Priorisierung der Merkmale anhand des Korrelationskoeffizienten  $r$ , wobei das obere Merkmal das wichtigste und das unterste das unwichtigste darstellt. Die Berechnungen wurden auf Grundlage des *PSY* Datensatzes bei  $t(n) = 250ms$  durchgeführt.

Die Auswertungen erfolgen aufbauend auf Abschnitt 5.3.2. Es werden zuerst die *FD*-Merkmale für die vier Datensätze *ALL*, *PSY*, *SPEC*, *SEL* berechnet, anschließend die Merkmale anhand des Korrelationskoeffizienten nach oben erläuterten Verfahren geordnet, so dass diese in einer Rangliste angeordnet werden. Um die Güte des Selektionsverfahrens zu ermitteln und somit eine Grundlage für einen Vergleich zu haben, werden unterschiedlich viele Merkmale der Rangliste (*subset*, Merkmalsatz) zum Anlernen eines Klassifikators genutzt. Die erreichte Klassifikationsrate des *subset* lässt dann im Vergleich zu anderen Verfahren eine Bewertung der *subset* hinsichtlich der Eignung für eine Klassifizierung zu.

#	FD-Merkmal (ungeordnet)	#	FD-r-Merkmal (geordnet)
1	Specific Loudness (Zeitverlauf) [32]	10	Specific Fluctuation Strength [8]
2	Specific Loudness (Zeitverlauf) [212]	8	Specific Fluctuation Strength [6]
3	Specific Loudness (Zeitverlauf) [227]	20	Specific Roughness [2]
4	Specific Loudness (Zeitverlauf) [239]	24	Specific Roughness [11]
5	Specific Loudness (Zeitverlauf) [240]	16	Specific Loudness [35]
6	Specific Roughness (Zeitverlauf) [4]	11	Specific Fluctuation Strength [9]
7	Specific Fluctuation Strength [5]	12	Specific Fluctuation Strength [10]
8	Specific Fluctuation Strength [6]	25	Lenkgeschwindigkeit
9	Specific Fluctuation Strength [7]	15	Specific Fluctuation Strength [21]
10	Specific Fluctuation Strength [8]	26	Lenkwinkel
11	Specific Fluctuation Strength [9]	22	Specific Roughness [4]
12	Specific Fluctuation Strength [10]	14	Specific Fluctuation Strength [16]
13	Specific Fluctuation Strength [15]	6	Specific Roughness (Zeitverlauf) [4]
14	Specific Fluctuation Strength [16]	27	Giergeschwindigkeit
15	Specific Fluctuation Strength [21]	19	Specific Loudness [239]
16	Specific Loudness [35]	3	Specific Loudness (Zeitverlauf) [227]
17	Specific Loudness [60]	4	Specific Loudness (Zeitverlauf) [239]
18	Specific Loudness [228]	17	Specific Loudness [60]
19	Specific Loudness [239]	2	Specific Loudness (Zeitverlauf) [212]
20	Specific Roughness [2]	9	Specific Fluctuation Strength [7]
21	Specific Roughness [3]	7	Specific Fluctuation Strength [5]
22	Specific Roughness [4]	21	Specific Roughness [3]
23	Specific Roughness [5]	1	Specific Loudness (Zeitverlauf) [32]
24	Specific Roughness [11]	23	Specific Roughness [5]
25	Lenkgeschwindigkeit	13	Specific Fluctuation Strength [15]
26	Lenkwinkel	18	Specific Loudness [228]
27	Giergeschwindigkeit	5	Specific Loudness (Zeitverlauf) [240]

Tabelle 5.10: Auflistung der Fisher-Diskriminante-Merkmale mit Korrelationskoeffizient-Rangordnung

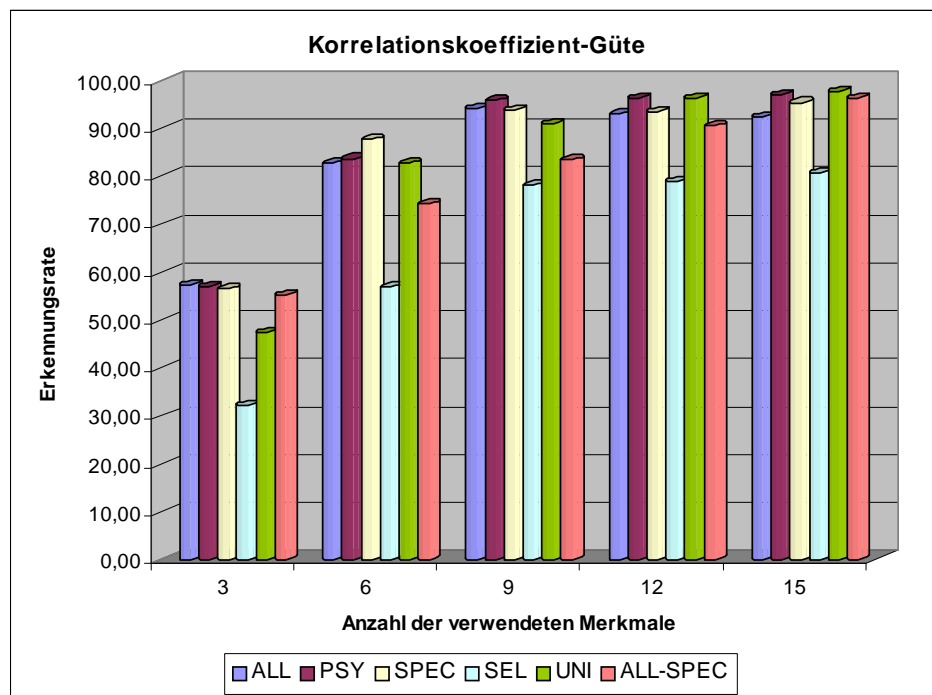


Abbildung 5.4: Vergleich der Korrelationskoeffizient-Klassifikationsraten

Aus Abbildung 5.4 (Analyse einer unterschiedlichen Anzahl verwendeter Merkmale des Korrelationskoeffizienten-subset für die vier Datensätze bei  $t(n) = 250ms$ ) wird ersichtlich, dass bereits ab sechs Merkmalen, je nach Datensatz, eine Erkennungsrate von über 80% erreicht wird. Ab zwölf Merkmalen wird



eine Rate größer 95% erreicht. Die einzige Ausnahme stellt der *SEL* Datensatz dar. Dieser besteht jedoch insgesamt aus nur 14 Einzelmerkmalen und erreicht damit ca. 75-80%, siehe Tabelle 5.8. Der *SEL* Datensatz stellt somit wieder den ungeeignetsten Datensatz dar. Für die *ALL*, *PSY*, *SPEC* Datensätze werden bei 15 verwendeten Merkmalen ebenfalls rund 95% Erkennungsrate erreicht. Betrachtet man den ungeordneten FD-Datensatz, so wurden ca. 98% für diese Datensätze erreicht, mit 39, 26 und 39 Merkmalen. Für den *ALL* und *SPEC* Datensatz bedeutet dies, dass auf Kosten von  $< 3\%$  Klassifikationsrate 27 Merkmale weniger berechnet werden müssen. Bei dem *PSY* Datensatz können bei ebenfalls  $< 3\%$  Einbuße 14 Merkmale weniger berechnet werden.

#### 5.3.4 Fisher-Diskriminante und Relief

Der Mehrklassen-*Relief*-Algorithmus aus Kapitel 4.3.5 priorisiert die Reihenfolge der verwendeten *Fisher*-Diskriminanten-Merkmale zur Mustererkennung um eine optimale Klassifikation bei geringer Anzahl verwendeter Merkmale realisieren zu können. In Tabelle 5.11 sind die ungeordneten Merkmale (der *Fisher*-Diskriminanten-Berechnung) und die durch den *Relief*-Algorithmus priorisierten Merkmale für den *PSY* Datensatz bei  $t(n) = 250ms$  exemplarisch aufgelistet, wobei das obere Merkmal das wichtigste und das unterste das unwichtigste darstellt.

#	FD-Merkmal (ungeordnet)	#	FD-Relief-Merkmal (geordnet)
1	Specific Loudness (Zeitverlauf) [32]	10	Specific Fluctuation Strength [8]
2	Specific Loudness (Zeitverlauf) [212]	9	Specific Fluctuation Strength [7]
3	Specific Loudness (Zeitverlauf) [227]	12	Specific Fluctuation Strength [10]
4	Specific Loudness (Zeitverlauf) [239]	11	Specific Fluctuation Strength [9]
5	Specific Loudness (Zeitverlauf) [240]	13	Specific Fluctuation Strength [15]
6	Specific Roughness (Zeitverlauf) [4]	14	Specific Fluctuation Strength [16]
7	Specific Fluctuation Strength [5]	7	Specific Fluctuation Strength [5]
8	Specific Fluctuation Strength [6]	8	Specific Fluctuation Strength [6]
9	Specific Fluctuation Strength [7]	26	Lenkwinkel
10	Specific Fluctuation Strength [8]	18	Specific Loudness [228]
11	Specific Fluctuation Strength [9]	19	Specific Loudness [239]
12	Specific Fluctuation Strength [10]	15	Specific Fluctuation Strength [21]
13	Specific Fluctuation Strength [15]	24	Specific Roughness [11]
14	Specific Fluctuation Strength [16]	22	Specific Roughness [4]
15	Specific Fluctuation Strength [21]	21	Specific Roughness [3]
16	Specific Loudness [35]	20	Specific Roughness [2]
17	Specific Loudness [60]	23	Specific Roughness [5]
18	Specific Loudness [228]	27	Giergeschwindigkeit
19	Specific Loudness [239]	6	Specific Roughness (Zeitverlauf) [4]
20	Specific Roughness [2]	17	Specific Loudness [60]
21	Specific Roughness [3]	16	Specific Loudness [35]
22	Specific Roughness [4]	1	Specific Loudness (Zeitverlauf) [32]
23	Specific Roughness [5]	2	Specific Loudness (Zeitverlauf) [212]
24	Specific Roughness [11]	3	Specific Loudness (Zeitverlauf) [227]
25	Lenkgeschwindigkeit	25	Lenkgeschwindigkeit
26	Lenkwinkel	4	Specific Loudness (Zeitverlauf) [239]
27	Giergeschwindigkeit	5	Specific Loudness (Zeitverlauf) [240]

Tabelle 5.11: Auflistung der *Fisher*-Diskriminante-Merkmale mit *Relief*-Rangordnung

Für jeden der vier Datensätze wurde eine solche Rangliste mit den FD-Merkmalen erstellt und anschließend der MLP-NN Klassifikator darauf trainiert, die zwölf Fehlerklassen zu unterscheiden. Die erreichte Erkennungsrate (Güte) wurde dann für eine verschiedenen große Anzahl verwendeter Merkmale berechnet; die Ergebnisse sind in Abbildung 5.5 bei unterschiedlicher Anzahl verwendeter

Merkmale des *Relief-subset* für die vier Datensätze bei  $t(n) = 250ms$  dargestellt.

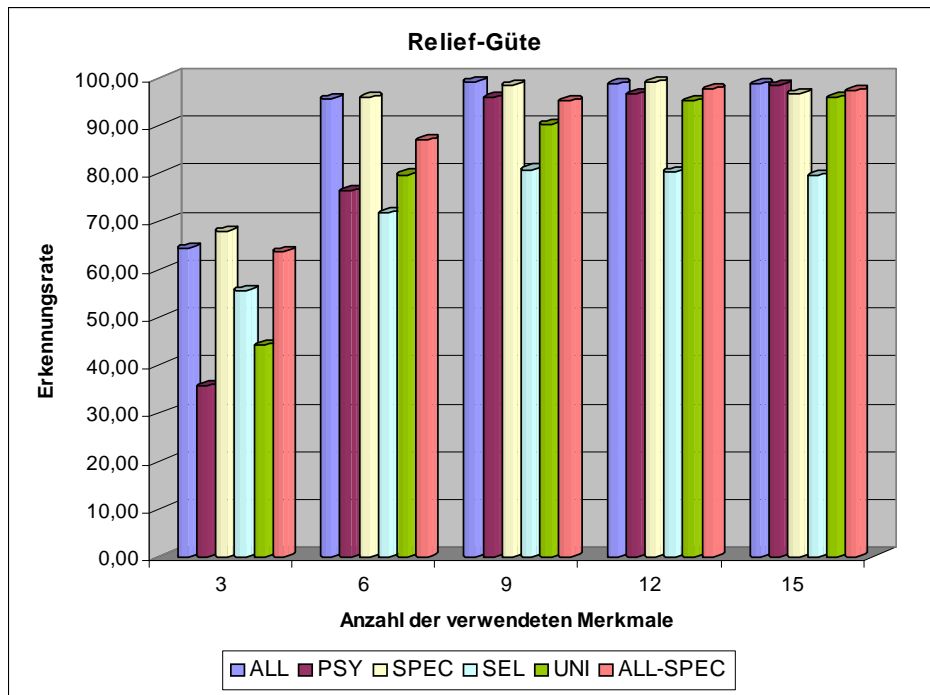


Abbildung 5.5: Vergleich der Relief-Klassifikationsraten

Es lässt sich wieder deutlich erkennen, dass der *SEL* Datensatz erneut die schlechteste Performance erreicht. Bereits bei 6 verwendeten Merkmalen erreichen die Datensätze *ALL*, *PSY* und *SPEC* Erkennungsraten von 80-85%, der *SEL* Datensatz hingegen nur 65%. Bereits bei 9 verwendeten Merkmalen werden Raten von  $> 90\%$  für die besten drei Datensätze erreicht. Ab 12 verwendeten Merkmalen liegen diese bereits über 95%, eine Verbesserung der Güte ist bei Erhöhung der verwendeten Merkmale nur noch im 1% Bereich wahrzunehmen.

Betrachtet man die Tabelle A.4 in Anhang A.1, so ist die Anordnung der mehrfach vorkommenden Merkmale in den unterschiedlichen Datensätzen (grüne Markierung) weniger gestreut im Vergleich zu Tabelle A.3.

### 5.3.5 Fisher-Diskriminante und Transinformation

Basierend auf der Transinformationsberechnung (*Mutual Information*, nichtlineares Verfahren, siehe Abschnitt 4.3.6.2), wird ein entwickelter Transinformations-Algorithmus auf die *Fisher*-Diskriminante-Merkmale angewendet, um eine Rangordnungsinformation zu erhalten.

Der *Mutual-Information*-Algorithmus basiert auf den Angaben in [PLD05]. Der Algorithmus lässt sich in zwei Bereiche einteilen:

- Berechnung der maximalen Transinformation zwischen der Merkmal- und Klasseninformation
- Berechnung der minimalen Redundanz anhand des Transinformations-Algorithmus' zwischen den Merkmalen

Tabelle 5.12 zeigt eine Priorisierung der Merkmale anhand des *Mutual-Information*-Algorithmus, wobei das obere Merkmal das wichtigste und das unterste das unwichtigste darstellt. Die Berechnungen wurden auf Grundlage des *PSY* Datensatzes bei  $t(n) = 250ms$  durchgeführt.

#	FD-Merkmal (ungeordnet)	#	FD-Mutual-Merkmal (geordnet)
1	Specific Loudness (Zeitverlauf) [32]	22	Specific Roughness [4]
2	Specific Loudness (Zeitverlauf) [212]	27	Giergeschwindigkeit
3	Specific Loudness (Zeitverlauf) [227]	13	Specific Fluctuation Strength [15]
4	Specific Loudness (Zeitverlauf) [239]	19	Specific Loudness [239]
5	Specific Loudness (Zeitverlauf) [240]	21	Specific Roughness [3]
6	Specific Roughness (Zeitverlauf) [4]	26	Lenkwinkel
7	Specific Fluctuation Strength [5]	15	Specific Fluctuation Strength [21]
8	Specific Fluctuation Strength [6]	23	Specific Roughness [5]
9	Specific Fluctuation Strength [7]	5	Specific Loudness (Zeitverlauf) [240]
10	Specific Fluctuation Strength [8]	16	Specific Loudness [35]
11	Specific Fluctuation Strength [9]	14	Specific Fluctuation Strength [16]
12	Specific Fluctuation Strength [10]	25	Lenkgeschwindigkeit
13	Specific Fluctuation Strength [15]	8	Specific Fluctuation Strength [6]
14	Specific Fluctuation Strength [16]	17	Specific Loudness [60]
15	Specific Fluctuation Strength [21]	2	Specific Loudness (Zeitverlauf) [212]
16	Specific Loudness [35]	1	Specific Loudness (Zeitverlauf) [32]
17	Specific Loudness [60]	4	Specific Loudness (Zeitverlauf) [239]
18	Specific Loudness [228]	20	Specific Roughness [2]
19	Specific Loudness [239]	6	Specific Roughness (Zeitverlauf) [4]
20	Specific Roughness [2]	24	Specific Roughness [11]
21	Specific Roughness [3]	7	Specific Fluctuation Strength [5]
22	Specific Roughness [4]	3	Specific Loudness (Zeitverlauf) [227]
23	Specific Roughness [5]	12	Specific Fluctuation Strength [10]
24	Specific Roughness [11]	18	Specific Loudness [228]
25	Lenkgeschwindigkeit	11	Specific Fluctuation Strength [9]
26	Lenkwinkel	9	Specific Fluctuation Strength [7]
27	Giergeschwindigkeit	10	Specific Fluctuation Strength [8]

Tabelle 5.12: Auflistung der *Fisher*-Diskriminante-Merkmale mit Transinformations-Rangordnung

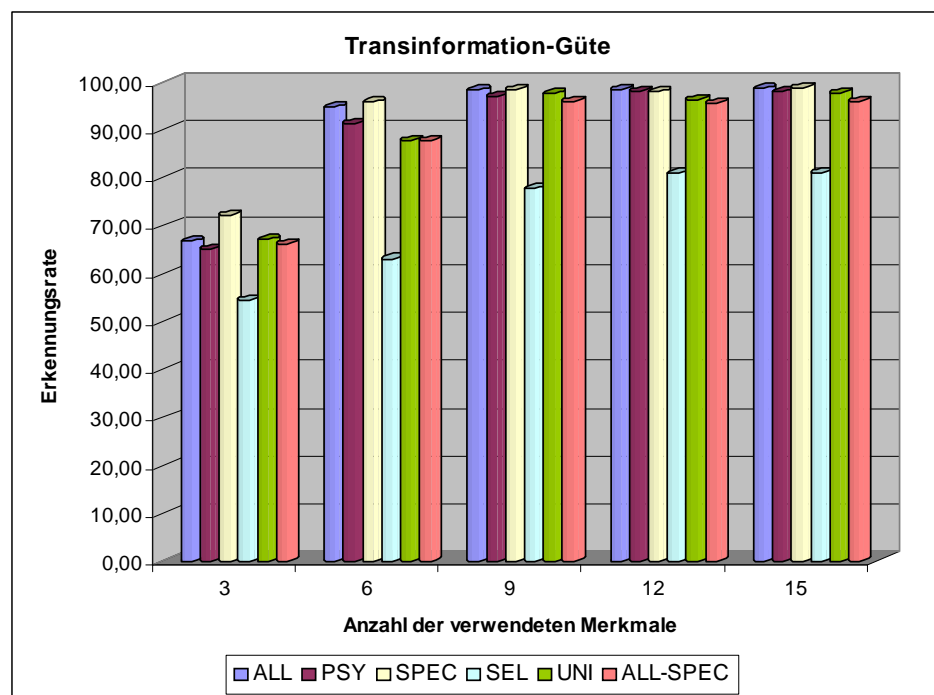


Abbildung 5.6: Vergleich der Transinformations-Klassifikationsraten

In Abbildung 5.6 ( $t(n) = 250ms$ ) ist bei unterschiedlicher Anzahl verwendeter Merkmale des Transinformations-*subset* ersichtlich, dass bereits bei 6 Merkmalen eine Klassifikationsrate größer 90 Prozent für alle Datensätze außer *SEL* erreicht wird. Mit 3 Merkmalen werden für diese Datensätze rund 65% Güte erreicht.

Im Vergleich der Spalten der Tabelle A.5 in Anhang A.1 ist außerdem erkennbar, dass der Transinformations-Algorithmus fast alle Merkmale, die in allen Datensätzen vorhanden sind (grüne Markierung), an oberster Stelle der Rangliste angeordnet werden.

### 5.3.6 Fisher-Diskriminante und Informationsgewinn

Der Informationsgewinn-Algorithmus (*Infogain*) (basierend auf Abschnitt 4.3.6.1) wird auf die *Fisher*-Diskriminante-Merkmale angewendet um eine Rangordnungsinformation zu erhalten. Der Informationsgewinn wird zum einen für jedes Merkmal zu allen anderen Merkmalen berechnet und zum anderen zwischen jedem Merkmal und der Klassenzuordnung. In Tabelle 5.13 erfolgte eine Priorisierung der Merkmale anhand des Informationsgewinn-Algorithmus, wobei das obere Merkmal das wichtigste und das unterste das unwichtigste darstellt. Die Berechnungen wurden auf Grundlage des *PSY* Datensatzes bei  $t(n) = 250ms$  durchgeführt.

#	FD-Merkmal (ungeordnet)	#	FD-Info-Merkmal (geordnet)
1	Specific Loudness (Zeitverlauf) [32]	5	Specific Loudness (Zeitverlauf) [240]
2	Specific Loudness (Zeitverlauf) [212]	1	Specific Loudness (Zeitverlauf) [32]
3	Specific Loudness (Zeitverlauf) [227]	20	Specific Roughness [2]
4	Specific Loudness (Zeitverlauf) [239]	19	Specific Loudness [239]
5	Specific Loudness (Zeitverlauf) [240]	16	Specific Loudness [35]
6	Specific Roughness (Zeitverlauf) [4]	14	Specific Fluctuation Strength [16]
7	Specific Fluctuation Strength [5]	2	Specific Loudness (Zeitverlauf) [212]
8	Specific Fluctuation Strength [6]	21	Specific Roughness [3]
9	Specific Fluctuation Strength [7]	17	Specific Loudness [60]
10	Specific Fluctuation Strength [8]	15	Specific Fluctuation Strength [21]
11	Specific Fluctuation Strength [9]	22	Specific Roughness [4]
12	Specific Fluctuation Strength [10]	18	Specific Loudness [228]
13	Specific Fluctuation Strength [15]	3	Specific Loudness (Zeitverlauf) [227]
14	Specific Fluctuation Strength [16]	4	Specific Loudness (Zeitverlauf) [239]
15	Specific Fluctuation Strength [21]	6	Specific Roughness (Zeitverlauf) [4]
16	Specific Loudness [35]	13	Specific Fluctuation Strength [15]
17	Specific Loudness [60]	7	Specific Fluctuation Strength [5]
18	Specific Loudness [228]	27	Giergeschwindigkeit
19	Specific Loudness [239]	10	Specific Fluctuation Strength [8]
20	Specific Roughness [2]	12	Specific Fluctuation Strength [10]
21	Specific Roughness [3]	25	Lenkgeschwindigkeit
22	Specific Roughness [4]	11	Specific Fluctuation Strength [9]
23	Specific Roughness [5]	26	Lenkwinkel
24	Specific Roughness [11]	24	Specific Roughness [11]
25	Lenkgeschwindigkeit	23	Specific Roughness [5]
26	Lenkwinkel	8	Specific Fluctuation Strength [6]
27	Giergeschwindigkeit	9	Specific Fluctuation Strength [7]

Tabelle 5.13: Auflistung der *Fisher*-Diskriminante-Merkmale mit Informationsgewinn-Rangordnung

In Abbildung 5.7 ist der Vergleich des Informationsgewinns bei unterschiedlicher Anzahl verwendeter Merkmale des *Infogain*-*subset* für die vier Datensätze bei  $t(n) = 250ms$  aufgezeigt. Am auffallendsten ist die durchweg bessere Güte für den Psychoakustik-Datensatz *PSY* bei einer Anzahl verwendeter Merkmale kleiner 9.

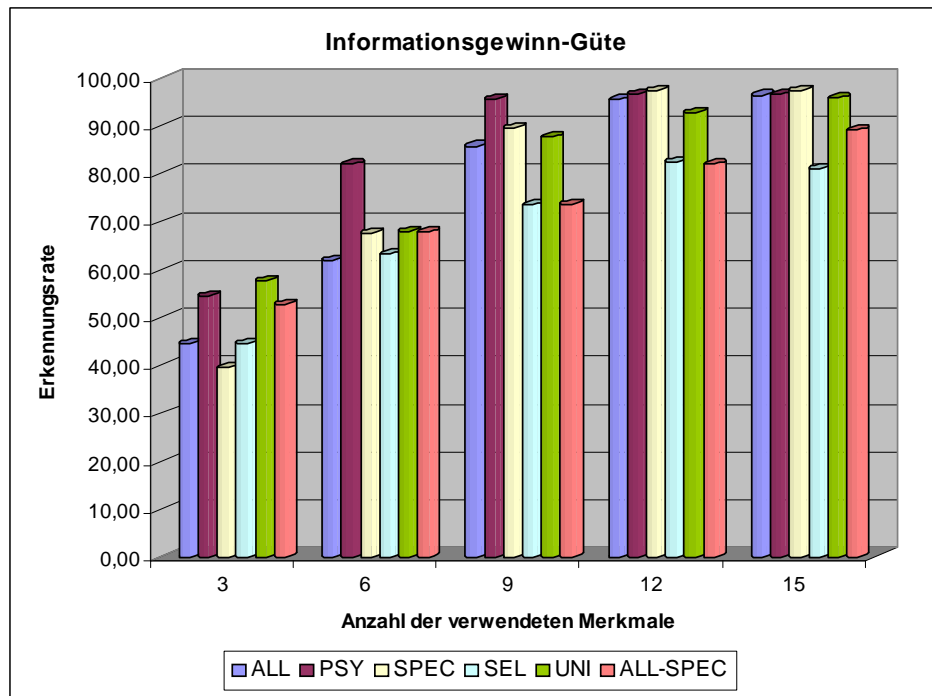


Abbildung 5.7: Vergleich der Informationsgewinn-Klassifikationsraten

Bei Verwendung von nur 3 Merkmalen wird mit dem Informationsgewinn eine Klassifikationsrate von rund 70 Prozent erreicht. Dabei werden laut Tabelle A.6 in Anhang A.1 die Merkmale „spezifische Lautheit“ (Band 239 und 38) und die „spezifische Schwankungsstärke“ (Band 16) für die Mustererkennung verwendet. Die grün markierten Merkmale der Tabelle A.6 sind im Vergleich sehr stark gestreut und nicht im oberen Bereich der Rangliste konzentriert.

### 5.3.7 Fisher-Diskriminante und FCBF

*Fast-Correlation-Based-Filtering* nach dem Verfahren von [YLG04], siehe Abschnitt 4.3.7, enthält zwei grundlegend angewendete Verfahren, die Berechnung der:

- *C-correlation*: Steht als Maß für die Korrelation zwischen jedem Merkmal und der Klasseninformation,
- *F-correlation*: Steht als Maß für die Korrelation zwischen jedem Paar der Merkmale.

Die sortierten Merkmale sind in Tabelle 5.14 für den *PSY* Datensatz abgebildet. Priorisierung der Merkmale erfolgte anhand des *FCBF*-Algorithmus, wobei das obere Merkmal das wichtigste und das unterste das unwichtigste darstellt. Die Berechnungen wurden auf Grundlage des *PSY* Datensatzes bei  $t(n) = 250ms$  durchgeführt.

Im Vergleich der Datensätze (siehe Abbildung 5.8, bei  $t(n) = 250ms$ ) zeigt sich, dass der *PSY* Datensatz in Verbindung mit dem *FCBF*-Verfahren bei Verwendung von weniger als 9 Merkmalen immer die höchsten Erkennungsraten erreicht. Bei 3 Merkmalen wird eine um ca. 20% bessere Erkennungsrate gegenüber dem *ALL* Datensatz erreicht.

#	FD-Merkmal (ungeordnet)	#	FD-FCBF-Merkmal (geordnet)
1	Specific Loudness (Zeitverlauf) [32]	14	Specific Fluctuation Strength [16]
2	Specific Loudness (Zeitverlauf) [212]	19	Specific Loudness [239]
3	Specific Loudness (Zeitverlauf) [227]	23	Specific Roughness [5]
4	Specific Loudness (Zeitverlauf) [239]	10	Specific Fluctuation Strength [8]
5	Specific Loudness (Zeitverlauf) [240]	9	Specific Fluctuation Strength [7]
6	Specific Roughness (Zeitverlauf) [4]	12	Specific Fluctuation Strength [10]
7	Specific Fluctuation Strength [5]	11	Specific Fluctuation Strength [9]
8	Specific Fluctuation Strength [6]	8	Specific Fluctuation Strength [6]
9	Specific Fluctuation Strength [7]	22	Specific Roughness [4]
10	Specific Fluctuation Strength [8]	18	Specific Loudness [228]
11	Specific Fluctuation Strength [9]	16	Specific Loudness [35]
12	Specific Fluctuation Strength [10]	13	Specific Fluctuation Strength [15]
13	Specific Fluctuation Strength [15]	7	Specific Fluctuation Strength [5]
14	Specific Fluctuation Strength [16]	20	Specific Roughness [2]
15	Specific Fluctuation Strength [21]	17	Specific Loudness [60]
16	Specific Loudness [35]	15	Specific Fluctuation Strength [21]
17	Specific Loudness [60]	21	Specific Roughness [3]
18	Specific Loudness [228]	1	Specific Loudness (Zeitverlauf) [32]
19	Specific Loudness [239]	2	Specific Loudness (Zeitverlauf) [212]
20	Specific Roughness [2]	3	Specific Loudness (Zeitverlauf) [227]
21	Specific Roughness [3]	4	Specific Loudness (Zeitverlauf) [239]
22	Specific Roughness [4]	5	Specific Loudness (Zeitverlauf) [240]
23	Specific Roughness [5]	6	Specific Roughness (Zeitverlauf) [4]
24	Specific Roughness [11]	24	Specific Roughness [11]
25	Lenkgeschwindigkeit	25	Lenkgeschwindigkeit
26	Lenkwinkel	26	Lenkwinkel
27	Giergeschwindigkeit	27	Giergeschwindigkeit

Tabelle 5.14: Auflistung der Fisher-Diskriminante-Merkmale mit FCBF-Rangordnung

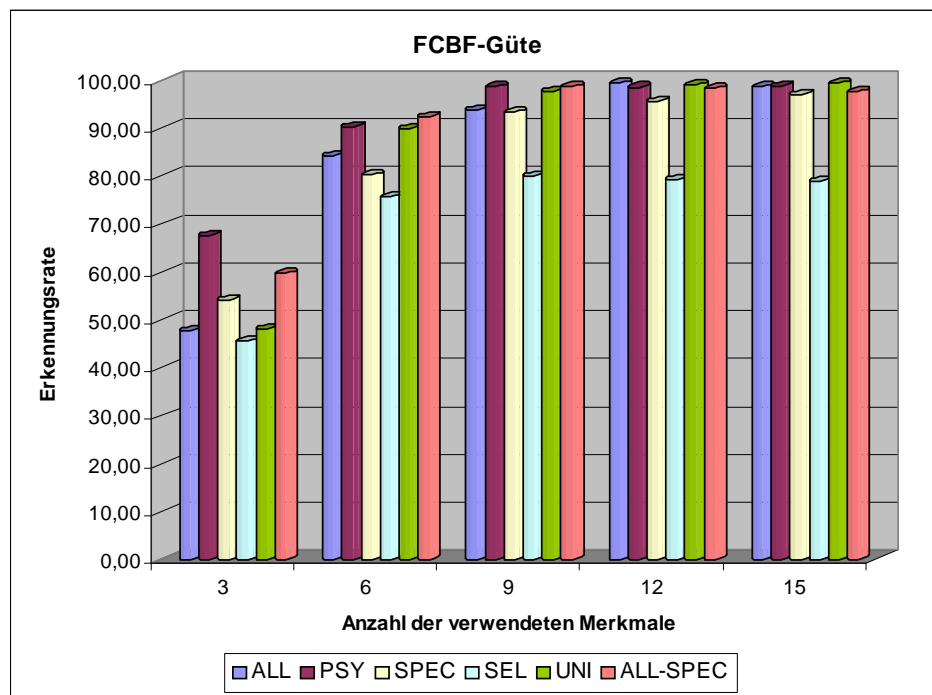


Abbildung 5.8: Vergleich der FCBF-Klassifikationsraten

Wiederum zeigt sich, dass bei Verwendung von mehr als 9 Merkmalen, alle Datensätze - abgesehen vom *SEL* Datensatz - eine ungefähr gleiche Erkennungsrate von ca. 96% aufweisen.

### 5.3.8 Fazit

Um abschließend die Performance der einzelnen Verfahren miteinander vergleichen zu können, wird ein Mittelwert der Klassifikationsrate über die vier verschiedenen Datensätze gebildet. Dieser ist in Abbildung 5.9 ersichtlich. Aus dieser kann eine Bewertung der mittleren Güte und allgemeiner Anwendbarkeit in Abhängigkeit von der verwendeten Anzahl von Merkmalen erfolgen. Wird eine größtmögliche Mustererkennungsrates bei geringster Anzahl von Merkmalen erwartet, so eignen sich hierfür besonders folgende Verfahren:

- Transinformation
- *Multi-Relief*
- *Fast-Correlation-Based-Filter*

Es fällt dabei auf, dass das beste Verfahren zur Merkmalsbewertung im Vergleich zu dem schlechtesten eine um mindestens 15% bessere Klassifikationsrate aufweist. Ab einer Merkmalanzahl höher 12 unterscheiden sich die Mittelwerte der einzelnen Reduktionsmethoden nur noch gering voneinander.

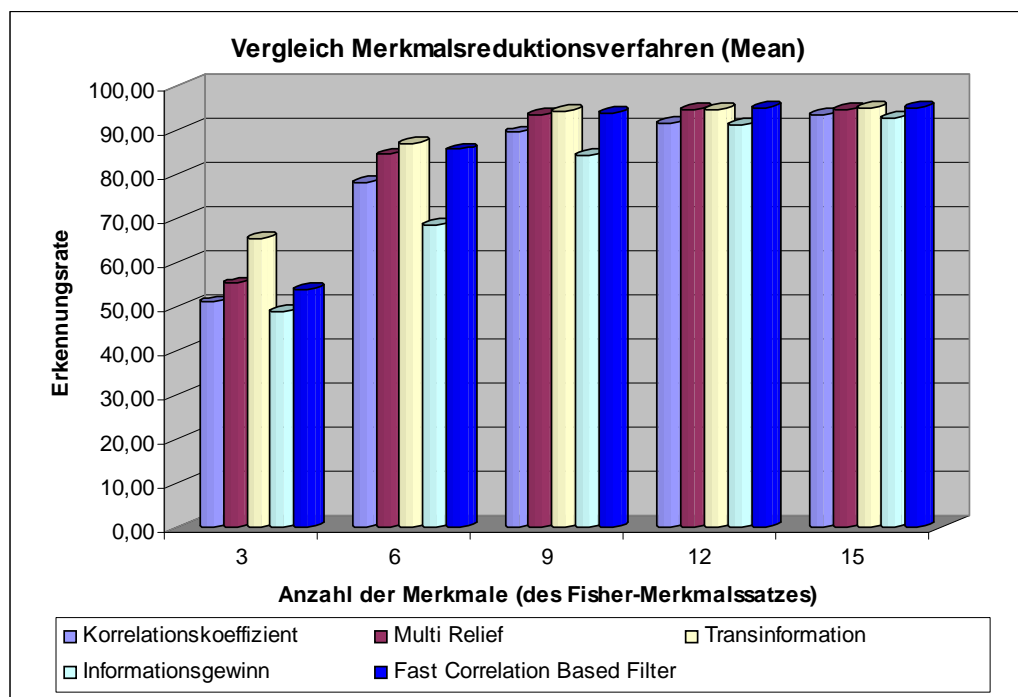


Abbildung 5.9: Merkmalreduktionsverfahren im Vergleich

Eine weitere Bewertung dieser Algorithmen erfolgt durch die Abschätzung des Rechenaufwands. Dafür wird auf einem Testrechner (*Intel Core 2 Duo CPU E6750 @2.66 GHz, 2 GB RAM*) die von Matlab® benötigte Rechenzeit für jeden Algorithmus bestimmt, siehe Tabelle 5.15. Dabei ist zu beachten, dass diese Werte nur für die realisierten Algorithmen gelten, jedoch stark von der Programmiersprache, den verwendeten *Compiler*-Funktionen und der Programmierweise abhängen. Es wird deutlich, dass die Berechnung des Korrelationskoeffizienten in 430 Millisekunden die

wenigste Rechenzeit benötigt. Für die drei im vorherigen Absatz erwähnten Merkmalreduktionsverfahren mit den besten Erkennungsraten (fett hervorgehoben) weist der *Multi-Relief*-Algorithmus die höchste Effizienz auf. Dieser benötigt mit 520 Millisekunden dennoch ca. 20 Prozent mehr Zeit als das Korrelationskoeffizient-Verfahren. Ähnliche Performance erreicht die Transinformation. Weit abgeschlagen ist der *FCBF*-Algorithmus, welcher mit 16,35 Sekunden ca. 38 mal soviel Rechenzeit benötigt wie das schnellste Verfahren.

Verfahren	mittlere Rechenzeit [s]	Faktor
Korrelationskoeffizient	0.43	1
<i>Multi-Relief</i>	0.52	1.2
Transinformation	0.55	1.3
Informationsgewinn	0.56	1.3
<i>FCBF</i>	16.35	38

Tabelle 5.15: Benötigte Rechenzeit der Merkmalreduktionsverfahren im Vergleich

Zusammenfassend zeigt sich, dass der *Multi-Relief*-Algorithmus die besten Ergebnisse im Bereich der Merkmalreduktion erreicht und im Vergleich zu den anderen Verfahren eine geringe Rechenzeit benötigt.

### 5.4 Fahrzeugspezifische akustische Fingerabdrücke im Vergleich

Für die vorangestellten Untersuchungen wurde ein Datensatz für den Fahrzeugtyp Polo erzeugt und klassifiziert. In einem weiteren Schritt wird ein Datensatz für den Fahrzeugtyp Scirocco und Polo erstellt, wobei beide die gleichen Klassen enthalten. Die Erstellung der Datensätze erfolgt analog zu den Beschreibungen in Kapitel 5.1. Im Folgenden wird der Datensatztyp *PSY* verwendet (psychoakustische Kenngrößen), mit einer Signalauschnittlänge  $t(n) = 250ms$ . Da nicht für alle Fahrzeuge in der Datenbank die gleichen Fehlerzustände vorliegen, ergaben sich für beide Fahrzeuge insgesamt sieben gemeinsame Klassen, welche in Tabelle 5.16 gezeigt sind.

Klasse	Klangbild/ Fehlerzustand
1	Koppelstange lose links
2	Koppelstange ausgeschlagen links
3	Koppelstange ausgeschlagen rechts
4	Radlager links
5	Radlager rechts
6	Gelenkwelle rechts
7	fehlerfrei/ Originalklang

Tabelle 5.16: Zustandsdefinitionen für einen Datensatz mit zwei verschiedenen Fahrzeugen

Folgende Fragestellungen, sollen im Rahmen der in diesem Abschnitt durchgeführten Untersuchungen beantwortet werden:

- Ist die Klassifikation des Polos bei einem auf den Scirocco angelernten Mustererkenner (oder umgedreht) möglich?



- Wie sind die Mustererkennungsraten, wenn ein gemeinsamer Datensatz für Training und Test verwendet wird?
- Gibt es Unterschiede/ Ähnlichkeiten in den Mustern der Fahrzeuge, bei gleicher Klasse?
- Sind die durch die Merkmalreduktionsverfahren bestimmten akustischen Fingerabdrücke (relevanten Merkmale) für beide Fahrzeugtypen gleich?

Im Endeffekt beschreiben alle diese Aspekte eine Analyse der Merkmale hinsichtlich Eignung für fahrzeugübergreifende akustische Mustererkennung. Um die Klassifizierbarkeit zu untersuchen, wurde der MLP-NN Klassifikator (siehe Klassifikatorenvergleich in Abschnitt 5.3.1) auf die fahrzeugspezifischen Datensätze angewendet. Dabei wurden 70% der Datensatz-Muster zum Anlernen und 30% zum Testen - nach dem *bootstrapping* Verfahren (siehe Abschnitt 4.4.1) - verwendet. Bei den Untersuchungen wurde die *z-Score*-Normierung nach Gleichung 5.1 in Abschnitt 5.3.1 genutzt. Dies bedeutet, dass der Mittelwert und die Streuung jeweils für jeden Datensatz neu berechnet werden müssen, da der Mittelwert und die Streuung der Merkmale je nach Fahrzeug variiert. In Tabelle 5.17 ist exemplarisch ein Vergleich von 13 Merkmalen für die Klasse 7 (fehlerfreier Zustand) dargestellt. Dabei wurde der Mittelwert (*mean*), das Minimum (*min*) und das Maximum (*max*) für die Merkmale der beiden Fahrzeuge Polo (*p*) und Scirocco (*s*) berechnet.

Merkmal	<i>mean<sub>p</sub></i>	<i>mean<sub>s</sub></i>	<i>min<sub>p</sub></i>	<i>min<sub>s</sub></i>	<i>max<sub>p</sub></i>	<i>max<sub>s</sub></i>	<i>std<sub>p</sub></i>	<i>std<sub>s</sub></i>
1	60.66	61.71	29.66	31.43	209.23	112.58	29.48	14.91
2	98.95	107.02	37.50	43.95	302.34	194.53	46.21	28.42
3	2.97	2.51	2.31	1.69	4.18	3.76	0.31	0.22
4	0.71	0.66	0.01	0.05	3.02	2.43	0.68	0.55
5	0.09	0.22	0.00	0.00	2.30	3.07	0.19	0.31
6	81.88	87.79	36.96	46.09	98.81	99.64	15.77	10.82
7	0.02	0.03	0.00	0.00	0.26	0.27	0.03	0.05
8	71.53	66.25	13.18	11.72	161.13	156.74	24.01	28.44
9	2.70	3.06	1.85	1.49	4.14	10.42	0.33	0.88
10	24.83	19.60	5.57	4.20	60.25	56.76	12.27	11.52
11	0.76	1.22	0.11	0.26	2.77	3.27	0.56	0.56
12	1.06	0.88	0.83	0.66	2.05	1.81	0.16	0.20
13	0.06	0.04	0.00	0.00	0.23	0.20	0.04	0.03

Tabelle 5.17: Exemplarischer Vergleich für Klasse 7, von 13 Merkmalen der Fahrzeuge Polo (*p*) und Scirocco (*s*) für die statistischen Kenngrößen Mittelwert (*mean*), Minimum (*min*) und Maximum (*max*)

In [Ruf09] wurden ebenfalls fahrzeugspezifische Muster verglichen und hinsichtlich Allgemeingültigkeit analysiert. Dabei fällt auf, dass [Ruf09] stark abweichende statistische Kenngrößen erhält, also die Wertebereiche der Merkmale für die zwei Fahrzeuge sehr stark voneinander abweichen. Da dieses Verhalten in den hier gezeigten Untersuchungen nicht ersichtlich ist, wird vermutet, dass bei [Ruf09] Skalierungs- oder Datensatzberechnungsfehler aufgetreten sind.

Die im Folgenden beschriebenen Erkennungsraten wurden über fünf Durchgänge gemittelt. In Tabelle 5.18 sind die gemittelten Erkennungsraten für die zwei Fahrzeuge bzw. daraus erzeugten drei Datensätze mit der entsprechenden Anzahl der Muster abgebildet. Es ist ersichtlich, dass der Polo am besten klassifiziert wird, die anderen beiden Datensätze jedoch nur um max. 3%

schlechtere Erkennungsraten generieren.

Datensatz	Erkennungsrate [%]	Anzahl d. Muster
Polo	99.56	2128
Scirocco	97.80	1456
Polo&Scirocco	96.84	3584

Tabelle 5.18: Erkennungsraten der Datensätze bei 9 Merkmalen über fünf Durchläufe gemittelt

Um diese sehr guten Erkennungsraten zu erreichen, wurde der *Relief*-Algorithmus (Abschnitt 4.3.5) zur Bewertung der Merkmale verwendet. Es sollen für jeden Datensatz die relevanten Merkmale gefunden und daraus der akustische Fingerabdruck gebildet werden. In Tabelle 5.19 erfolgt der Vergleich der berechneten Fingerabdrücke, und die *Fisher*-Diskriminanten der Merkmale sind ebenfalls abgebildet. Die *Fisher*-Diskriminante (FD) dient als Faktor zur Bewertung der Trennbarkeit des Merkmals gegenüber den 7 Klassen.

Im nächsten Schritt steht die Überprüfung der Allgemeingültigkeit der Merkmal-*subsets* bzw. akustischen Fingerabdrücke im Blickpunkt. Aus Tabelle 5.19 geht hervor, dass bestimmte Merkmale in allen Datensätzen vorkommen. Dabei ist jedoch auch anhand der *Fisher*-Diskriminante ersichtlich, dass diese Merkmale nicht in jedem Datensatz den gleichen Beitrag zur Trennung der Merkmale beitragen. Zum Beispiel hat das Merkmal *Specific Roughness* [4] einen FD-Wert von 1.42 für den Polo Datensatz, was einer mittleren Trennbarkeit entspricht. Das gleiche Merkmal hat jedoch einen FD-Wert von 6.21 im Scirocco Datensatz, was einer sehr guten Trennbarkeit entspricht.

Es fällt also auf, dass jeder Datensatz einen eigenen akustischen Fingerabdruck aufweist, um ein optimales Klassifikationsergebnis zu erreichen. Diese Erkenntnis bedeutet, dass jeder Datensatz einen speziellen Fingerabdruck hat. Um die Allgemeingültigkeit zu untersuchen, wurden die akustischen Fingerabdrücke eines Datensatzes auch auf die anderen Datensätze angewendet, siehe Tabelle 5.20.

akustischer Fingerabdruck	Datensatz	Erkennungsrate [%]
Polo	Scirocco	97.4
	Polo&Scirocco	95.13
Scirocco	Polo	98.09
	Polo&Scirocco	91.65
Polo&Scirocco	Polo	98.90
	Scirocco	98.58

Tabelle 5.20: Untersuchung zur Allgemeingültigkeit der akustischen Fingerabdrücke

In der linken Spalte sind die verwendeten Merkmal-*subsets* (siehe Tabelle 5.19) durch den Fahrzeugnamen spezifiziert. Anschließend wird anhand der definierten Merkmale für den jeweiligen Datensatz die Erkennungsrate berechnet. Es wird deutlich, dass mit allen drei Fingerabdrücken sehr gute Ergebnisse von über 90% erreicht werden. Die schlechteste Erkennungsrate von 91.65% wurde für den Fall, dass der Scirocco-Fingerabdruck zum Anlernen und Testen des Polo&Scirocco-Datensatzes verwendet wird, erreicht. Wird der gleiche Fingerabdruck jedoch auf

#	Polo	FD	Scirocco	FD	Polo&Scirocco	FD
1	Specific Fluctuation Strength [7]	4.20	Specific Fluctuation Strength [16]	1.88	Specific Fluctuation Strength [8]	1.86
2	Specific Fluctuation Strength [15]	3.73	Specific Fluctuation Strength [10]	6.74	Specific Fluctuation Strength [10]	2.89
3	Specific Fluctuation Strength [10]	3.29	Specific Fluctuation Strength [11]	10.23	Specific Fluctuation Strength [9]	3.85
4	Specific Fluctuation Strength [16]	2.11	Specific Fluctuation Strength [4]	1.55	Specific Fluctuation Strength [15]	2.08
5	Specific Fluctuation Strength [6]	1.37	Specific Fluctuation Strength [5]	0.68	Specific Fluctuation Strength [5]	0.87
6	Specific Roughness [4]	1.42	Specific Roughness [4]	6.21	Specific Fluctuation Strength [6]	1.12
7	Specific Loudness [31]	2.95	Specific Roughness [3]	7.09	Specific Roughness [3]	0.40
8	Specific Fluctuation Strength [21]	1.14	Specific Roughness [5]	5.47	Specific Roughness [5]	2.83
9	Specific Fluctuation Strength [20]	1.12	Specific Roughness [9]	1.07	Specific Fluctuation Strength [21]	2.55

Tabelle 5.19: Die relevanten Merkmale - akustischen Fingerabdrücke - für die drei generierten Datensätze wurden mit dem *Relief*-Algorithmus ermittelt.

den Polo-Datensatz angewandt, werden nahezu die Werte des Fingerabdruck-spezifischen Datensatzes erreicht. Betrachtet man äquivalent dazu den Fall, dass der Polo-Fingerabdruck zum Anlernen und Testen verwendet wird, ist die gleiche Tendenz ersichtlich. Es zeigt sich deutlich, dass die berechneten akustischen Fingerabdrücke sich ähneln und sehr gute Erkennungsraten liefern, egal auf welchen Datensatz diese angewendet werden.

Die Normierung jedes einzelnen Merkmals erfolgte wie beschrieben durch Subtraktion des Mittelwerts und Division durch die Standardabweichung. Die Normierung auf die Standardnormalverteilung setzt jedoch voraus, dass die Statistik der Datensätze/ Merkmale bekannt ist. Für das in diesem Abschnitt vorgegebene Problem wird jedoch deutlich, dass auch Änderungen in den Merkmalen auftreten, wenn sich der Fahrzeugtyp verändert. In Tabelle 5.17 sind die statistischen Kenngrößen der zwei Fahrzeuge für 13 Merkmale aus dem *SEL*-Datensatz (ohne fahrzeugbeschreibende CAN-Kenngrößen) dargestellt. Die Änderung der Erkennungsraten in Abhängigkeit des verwendeten akustischen Fingerabdrucks ist darauf zurückzuführen, dass je nach verwendeten Merkmalen unterschiedliche fahrzeugabhängige Ausprägungen der Merkmale auftreten. So kann der RMS-Wert (Merkmal 4) zwischen den Fahrzeugen sehr ähnlich sein (für den Polo 0.71 und für den Scirocco 0.66), aber die Lautheit (Merkmal 10) sehr unterschiedlich (24.83 und 19.60) sein. Ist der RMS-Wert Teil des Fingerabdrucks, so wird dieser nahezu gleichen Einfluss auf alle Datensätze haben. Der *Loudness*-Wert hingegen ist nur schlecht zur Klassifikation eines fahrzeugübergreifenden Datensatzes geeignet.

## 6 Agent zur Fehlerindikation

Als Diagnoseglied zur Erkennung, ob überhaupt eine komplexe Merkmalberechnung, -reduktion und Klassifikation ausgeführt werden soll, dient der IAD-Agent, wobei IAD für "Intelligente Akustikdiagnose" steht. Dieser ist ein autonomer Indikator, welcher angibt, ob ein Fehlerzustand vorliegt oder nicht. Als Eingangsgröße dient dabei das Mikrofonsignal aus dem Fahrzeuginnenraum. In diesem Kapitel wird unter Abschnitt 6.1 eine Methode zur Trennung verschiedener Geräusche (des Mikrofonsignals) basierend auf Korrelationsanalyse vorgestellt. Diese Methode stellt jedoch kein stabiles<sup>1</sup> System zu Mikrofonsignalanalyse dar, sondern dient viel mehr als grundlegende Einführung in die Interpretation der Merkmale, welche aus dem Mikrofonsignal berechnet werden. Im darauf folgenden Abschnitt 6.2 wird das aus der Konzeptphase und den vorgelagerten Untersuchungen erzeugte Modell zur Erkennung von Fehlergeräuschen anhand eines Mikrofons im Fahrzeuginnenraum - basierend auf selbstorganisierenden Karten - vorgestellt; dem so genannten SOM-Agent.

### 6.1 Korrelationsmethode zur Trennung akustischer Zustände

#### 6.1.1 Einführung

Eine objektive Trennung von auffälligen Geräuschen, die sich vorwiegend in der Fahrzeuginnenraum-Akustik bemerkbar machen, ist Bestandteil dieser hier vorgestellten Methode. Einzelkomponenten oder ganze Baugruppen des Fahrzeugs können eine Vielzahl unterschiedlicher Geräuscharten verursachen. Bisher werden diese Probleme meist dadurch gelöst, dass ein Sachverständiger eine Probefahrt macht und eine subjektive Bewertung abgibt. Diese Bewertung ist von vielen Faktoren abhängig und wird stark von den Erfahrungswerten des Prüfers beeinflusst. Ziel der hier vorgestellten Methode ist es festzustellen, ob es möglich ist, anhand von diversen Innenraummessungen die verschiedenen Geräuscharten bzw. Zustände, auffälliges oder normales/ unauffälliges Geräusch, anhand des Korrelationskoeffizienten voneinander zu trennen. Dazu ist es notwendig, das Zeitsignal der Messmikrofone in mathematisch beschreibbare Kenngrößen beziehungsweise Parameter zu zerlegen. Die Erkennung des linearen Zusammenhangs zwischen jedem dieser akustischen Parameter und im Weiteren zwischen den Parametern und den Zuständen sind Hauptbestandteil der Korrelationskoeffizienten-Analyse. Diese gehört zu den Filter-Methoden [Cha05], wie z.B. auch die *Fisher-Diskriminante*, das *Fast-Correlation-Based-Filter* und der *Relief-Algorithmus* [BD05], [Yu203]. Diese Algorithmen werden zur Dimensionsreduktion und zur Bestimmung der Relevanz der

---

<sup>1</sup>Stabil steht hierbei im Sinne von unabhängig gegenüber Störungen/ Umwelteinflüssen von außen und der damit verbundenen statistischen Sicherheit der generierten Modellausgabe.

verwendeten Parameter (bzw. auch als Merkmale bezeichnet) genutzt. Gegenüber den Wrapper-Methoden [Cha05] sind diese Filtermethoden weniger aufwendig und mit beliebigen Klassifikatoren anwendbar [Nie88], erreichen jedoch meist nicht so eine hohe Genauigkeit [HS97]. In der hier vorgestellten Methode soll der Korrelationskoeffizient nicht nur zur Bewertung der Parameter, sondern auch zur Unterscheidung zweier akustischer Zustände eingesetzt werden. Im Folgenden wird diese Methode zur objektiven Trennung verschiedener Geräuscharten bei einem VW Polo angewendet, die Methodik erläutert und die Ergebnisse interpretiert.

### 6.1.2 Modellentwurf

Der Korrelationskoeffizient stellt eine mathematische Größe zur Beurteilung des linearen Zusammenhangs zwischen zwei Parametern dar. Parameter eines akustischen Signals können z.B. der Effektivwert, spektrale Werte oder Transformationskoeffizienten sein. Der Experte definiert verschiedene Zustände. Prinzipiell lassen sich immer zwei definieren:

- Zustand 1: normales Geräusch,
- Zustand 2: auffälliges Geräusch (welches anschließend einer Geräuschart zugeordnet wird).

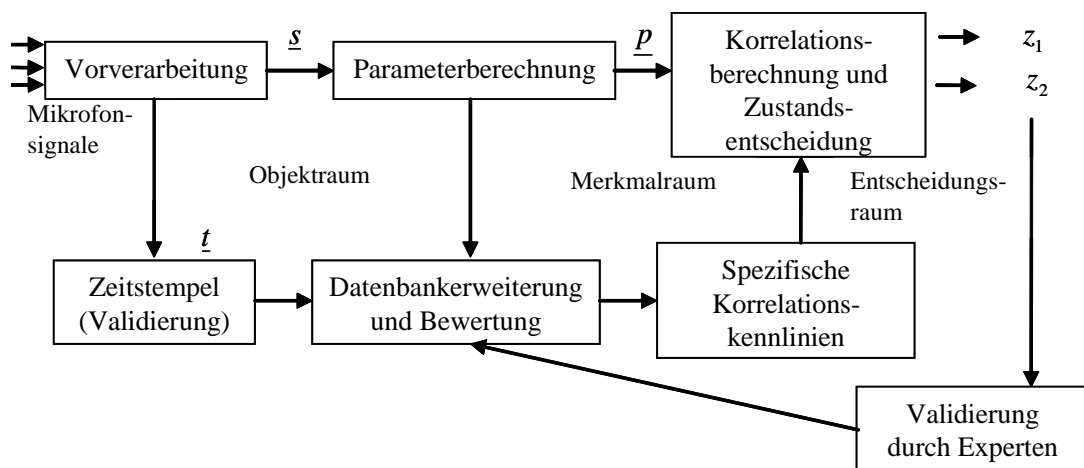


Abbildung 6.1: Darstellung der Geräuschunterscheidung anhand des Korrelationskoeffizienten (Quelle: [TOB09])

Im ersten Schritt kann durch eine Korrelationskoeffizientenbestimmung zwischen den Zuständen und den dafür berechneten Parametern eine Relevanz abgeleitet werden, vorausgesetzt es besteht ein linearer Zusammenhang zwischen der Geräuschänderung und den berechneten Parametern. Die Parameter werden nach Relevanz angeordnet, auf dessen Basis eine Parameterreduktion erfolgt (irrelevante Parameter werden dadurch vernachlässigt). Betrachtet man nun den Korrelationskoeffizienten von jedem einzelnen Parameter zu dem Parametersatz, so bildet sich eine Kennlinie, welche den linearen Zusammenhang eines jeden Parameters zu den anderen widerspiegelt, in Abhängigkeit des vorliegenden Zustandes. Ändert sich der Zustand, so ändert sich auch der lineare Zusammenhang zwischen den Parametern. Die Kennlinie ist somit repräsentativ für den Zustand. Auf Grundlage einer Datenbank kann somit eine Zuordnung eines unbekannten Parametersatzes zu einem Zustand erfolgen. In Abbildung 6.1 ist das Modellschema abgebildet. Der

Bereich "Vorverarbeitung" umfasst die Aufnahme des Innenraumgeräusches, sowie die Markierung der normalen und auffälligen Abschnitte. Im Modul "Parameterberechnung" werden ausgewählte akustische Parameter berechnet bzw. normiert/ skaliert, und zusätzliche Fahrzeugdaten werden vom Fahrzeug-CAN importiert. Die weiteren Schritte werden im Folgenden beschrieben.

### 6.1.2.1 Vorverarbeitung und Datenbank

Der Zeitstempel enthält hier zwecks Validierung zunächst eine manuelle Zuordnung durch einen Experten, in welchen Zeitbereichen Auffälligkeiten vorhanden sind. Der Bereich "Datenbankerweiterung und Bewertung" repräsentiert eine akustische Datenbank, welche die Parameter, den Zeitstempel und die Information über den Zustand (Geräuschart) von jedem Datensatz enthält. Die Ablage der Parameter, des Zeitstempels (wenn vorhanden) und des Zustandes in einem Datensatz erfolgt jeweils nach einer festen Zeitkonstante  $t$ , welche 250 ms beträgt. Aus einer Messung können also auch mehrere Datensätze erstellt werden. Des Weiteren erfolgt die Anwendung des Korrelationskoeffizienten zum Erstellen einer Rangliste, welche für jeden Parameter die Relevanz, bezogen auf den Zustand widerspiegelt. Neue Datensätze werden in der Datenbank aufgenommen, wenn der Zustand dafür bekannt ist. Die Zustandsentscheidung des Blockes „Korrelationsberechnung und Zustandsentscheidung“ in Abbildung 6.1 muss also vom Experten bestätigt werden.

### 6.1.2.2 Korrelationskennlinien

Zur Bereitstellung der spezifischen Korrelationskennlinien  $R_z$  wird eine Anzahl  $P$  von Parametern definiert, welche in den Beurteilungsprozess involviert werden sollen. Anschließend erfolgt die Berechnung des Korrelationskoeffizienten zwischen jedem in der Datenbank vorhandenen Zustand und jedem Parameter. Als Basis für die Darstellung des Korrelationskoeffizienten in Abhängigkeit von der Parameterkombination wird eine Kennlinie gebildet. Es ergeben sich somit für jeden Zustand  $P$ -Kennlinien, und jede Kennlinie zeigt den Korrelationskoeffizienten des Parameters  $p_i$  mit den Parametern  $p_1 \dots P$ , siehe Abbildung 6.2. Diese zeigt die zustandsspezifischen Korrelationskennlinien R1 des Zustand 1 (auffälliges Geräusch) und R2 des Zustands 2 (unauffälliges Geräusch), für den Parameter 6. Die Korrelationsberechnung und Zustandsentscheidung erfolgt für die aktuellen Mikrofonsignale. In Abhängigkeit von  $t$  steht eine unterschiedliche Anzahl von Datensätzen zur Verfügung, für die im Zeitstempel eine Auffälligkeit signalisiert wird. Als Ergebnis entstehen  $P$  Kennlinien  $R_x$ , welche den Zustand der Auffälligkeit repräsentieren. Die Zuordnung des aktuellen Datensatzes zu einem Zustand erfolgt anhand des kleinsten mittleren quadratischen Fehlers zwischen dem Kennliniensatz des aktuellen Mikrofonsignals und den spezifischen Korrelationskennlinien.

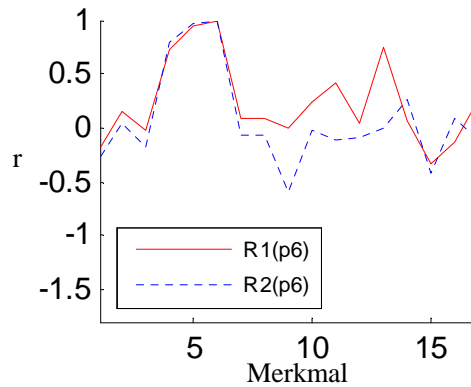


Abbildung 6.2: Zustandsspezifische Korrelationskennlinien (Quelle: [TOB09])

### 6.1.2.3 Mathematische Grundlagen

In Abschnitt 4.3.4 ist die Berechnung des Korrelationskoeffizienten gegeben. Bezogen auf die möglichen Kenngrößenkombinationen wird der Korrelationskoeffizient als Kennlinie abgebildet. Die zwei für jedes Merkmal berechneten zustandsspezifischen Korrelationskennlinien  $R_z$  sollen den Inhalt der Datenbank beschreiben. Die Datensatzberechnung erfolgt in Abhängigkeit der Zeitkonstanten von 250 Millisekunden, weshalb für jedes Mikrofonsignal mehrere Datensätze berechnet werden, wobei ein Datensatz aus den berechneten Parametern, dem Zeitstempel und der Zustandsinformation besteht. Die aus den aktuellen Mikrofonsignalen erzeugten Datensätze werden anschließend im Block „Korrelationsberechnung und Zustandsentscheidung“ verarbeitet. Für die Beurteilung des Verfahrens hinsichtlich objektiver Unterscheidbarkeit werden die Datensätze nach dem Zustand anhand des Zeitstempels getrennt und jeweils die Korrelationskennlinie  $R_x$  (analog zu der Berechnung der spezifischen Korrelationskennlinien  $R_z$ ) berechnet. Anschließend wird der mittlere quadratische Fehler  $\bar{e}$  (mean-squared-error, MSE) für jedes Merkmal aus der berechneten Kennlinien und den spezifischen Korrelationskennlinie wie folgt berechnet:

$$\bar{e}(p_i, z) = \frac{1}{P} \sum_{i=1}^P (R_z(p_i) - R_x(p_i))^2 \quad (6.1)$$

Dabei definiert  $z$  den Zustand (in diesem Beispiel: 1 oder 2). Der MSE ist hoch, wenn die spezifische Korrelationskennlinie und die aktuelle Korrelationskennlinie sich nicht ähneln. Nähern sich die zwei Kennlinien aneinander an, so nimmt der MSE ab. Bei kleinem MSE weisen die Parameter  $p_i$  der aktuellen Korrelationskennlinie  $R_x$  also ungefähr den gleichen Zusammenhang auf wie die ausgewählte spezifische Korrelationskennlinie  $R_z$ .

Berechnet man also für jedes  $p_i$  das MSE-Minimum, so kann dadurch eine Zuordnung zu den Zuständen wie folgt erreicht werden:

$$Z = \frac{1}{P} \sum_{i=1}^P \min [\bar{e}(z)] \quad (6.2)$$



wobei  $z = [1; 2]$  den ausgewählten Zustand und  $\vec{e}(z) = [\bar{e}(p_1, z) \dots \bar{e}(p_P, z)]$  die vom Zustand abhängigen MSE repräsentiert. Wird durch dieses Verfahren eine Trennbarkeit dieser zwei Zustände durch die Korrelationskennlinien nachgewiesen, so kann dieses auch bei einer durchgängigen Untersuchung des gesamten Mikrofonsignals erwartet werden. Dabei wird die Korrelationskennlinie fortlaufend berechnet und mit der Datenbank abgeglichen.

### 6.1.3 Algorithmus

Auf den Parametern, welche in der Datenbank mit den bekannten Zuständen gespeichert sind, wird zuerst der Algorithmus 6.1 ausgeführt. Von Zeile 1 bis 7 findet die Priorisierung der Parameter anhand ihrer Klassenkorrelation statt. Die Funktion „CorrCoeff“ stellt die Formel 4.12 dar. Ab Zeile 9 bis 18 werden die spezifischen Korrelationskennlinien für die in der Datenbank definierten Zustände berechnet. Dabei werden durch die Funktion „getNextElements“ die Parameter aus der Datenbank extrahiert, welche dem aktuellen Zustand zur Berechnung der Kennlinien angehören.

---

**Algorithmus 6.1** Zustandskorrelation und spezifische Korrelationskennlinien

---

*input:*

P: der Parametersatz mit den Parametern  $P_1, P_2, \dots, P_M$

C: Die Zustandszugehörigkeit

N: Die Anzahl der Zustände

Data(P,C): the data set

*output:*

Rlist: Die spezifischen Korrelationskennlinien

```
1 Plist=NULL
2 for i=1 to M do
3     calculate CorrCoeff(P(i),C) for P(i)
4     append to Plist
5 end
6 order Plist in descending abs(CorrCoeff(P(i),C)) value
7 sort P in order of Plist
8
9 Rlist=NULL
10 for c=1 to N do
11     Pc=getNextElements(P,c)
12     for i=1 to M do
13         for j=1 to M do
14             calculate CorrCoeff(Pc(i),Pc(j)) for Pc(i), Pc(j)
15             append to Rlist(c)
16         end
17     end
18 end
```

---

In „Rlist“ werden die zustandsabhängigen spezifischen Korrelationskennlinien abgespeichert. Diese dienen als Grundlage für die Zustandszuordnung eines unbewerteten Parametersatzes; die Vorgehensweise zur Berechnung der Korrelationskennlinien, des MSE und die Zuordnung eines Zustands ist in Algorithmus 6.2 gegeben. Von Zeile 1 bis 9 werden die Korrelationskennlinien für

die Parametersätze des aktuell zu untersuchenden Mikrofonsignals auf Basis der von der „rand“-Funktion definierten Parametersätze berechnet. Diese wählt stochastisch in Abhängigkeit von Faktor  $D$  (Anzahl zu verwendender Parametersätze) und  $\Delta D$  (Anzahl zur Verfügung stehender Parametersätze) die relevanten Parametersätze aus. Als Ergebnis werden die Kennlinien in „R“ zurückgegeben. Der MSE wird aus der gemittelten Differenz von „Rlist“ und „R“ berechnet und in „MSE(c)“, also in Abhängigkeit des gerade untersuchten Zustandes „c“ bzw. der verwendeten spezifischen Korrelationskennlinie berechnet, siehe Zeile 10 bis 18. In Zeile 19 wird abschließend der minimale Fehler in „MSE(c)“ gesucht, welcher den resultierenden Zustand für das Mikrofonsignal darstellt.

---

### Algorithmus 6.2 Korrelationskennlinien und Mean-Squared-Error

---

*input:*

P: der Parametersatz mit den Parametern  $P_1, P_2, \dots, P_M$

N: Die Anzahl der spezifischen Korrelationskennlinien

Rlist: Die spezifischen Korrelationskennlinien der Datenbank

D: Anzahl verwendeter Parametersätze

dD: Anzahl zur Verfügung stehender Parametersätze

*output:*

Z: Der berechnete Zustand

```
1 R=NULL
2 S=rand(D,dD)
3 for i=1 to M do
4     for j=1 to M do
5         calculate CorrCoeff(P(S,i),P(S,j)) for P(S,i), P(S,j)
6         append to R
7     end
8 end
9
10 E=NULL
11 MSE=NULL
12 for i=1 to M do
13     for c=1 to N do
14         calculate (Rlist(i,c)-R(i))^2 for Rlist(i,c), R(i)
15         append to E(c)
16     end
17     MSE(i,c)=mean(E(c))
18 end
19 Z=min(MSE(i,c))
```

---

#### 6.1.4 Test des Verfahrens und Auswertung

Im ersten Schritt steht eine Datenbank mit insgesamt max. ~ 2400 Datensätzen (ca. 600sec) zur Verfügung. Untersucht wird hierbei auch der Einfluss der Datenbankgröße  $N_{DB}$  auf die Zustandserkennung. Dabei fällt auf, dass ab einer Datenbankgröße kleiner 30 Prozent (720 Datensätze) die Erkennungsraten der beiden Zustände stärker voneinander abweichen und der Zustand 1 „auffälliges Geräusch“ eine ca. 10% bessere Erkennungsrate aufweist als der Zustand 2. Inhalt

der Datenbank sind 12 verschiedene Geräuscharten, wobei 11 davon als „auffälliges Geräusch“ deklariert werden und eins das „unauffällige Geräusch“ darstellt. Für jeden Zustand entstammen die berechneten Datensätze aus verschiedenen Fahrsituationen (z.B. Fahrbahnbelag, Geschwindigkeit) und variierenden Umweltbedingungen (z.B. Temperatur, Feuchtigkeit). Jedem Datensatz wurde ein akustischer Zustand durch einen Experten zugeordnet und die Datensätze berechnet. Auf Basis der bewerteten Datensätze erfolgte die Ermittlung der Relevanz der Parameter mit der Merkmal-/ Klassen-Korrelation (Tabelle 6.1). Die ungeordneten Parameter sind auf der linken Seite und die anhand der Parameter-/ Klassen-Korrelation berechnete Rangfolge ist auf der rechten Seite abgebildet. Umso niedriger der Rang, desto weniger besteht ein linearer Zusammenhang zwischen der Parameter- und Zustandsänderung.

#	Parameter (ungeordnet)	#	Parameter (geordnet)
1	Spectral Centroid	12	Sharpness (Time)
2	Spectral Rolloff	8	Modulation Frequency (Time)
3	Crestfactor	11	Roughness (Time)
4	RMS	3	Crestfactor
5	Spectral Fluctuation	1	Spectral Centroid
6	Articulation Index (Time)	2	Spectral Rolloff
7	Fluctuation Strength (Time)	7	Fluctuation Strength (Time)
8	Modulation Frequency (Time)	10	Loudness (Time)
9	Kurtosis (Time)	16	Speed
10	Loudness (Time)	9	Kurtosis (Time)
11	Roughness (Time)	17	Yaw Rate
12	Sharpness (Time)	14	Steering Speed
13	Tonality (Time)	6	Articulation Index (Time)
14	Steering Speed	15	Steering Angle
15	Steering Angle	5	Spectral Fluctuation
16	Speed	13	Tonality (Time)
17	Yaw Rate	4	RMS

Tabelle 6.1: Rangordnung anhand der Parameter-/ Klassen-Korrelation

Es ist deutlich sichtbar, dass der Effektivwert (niedrigster Rang) des Mikrofonsignals einen deutlich schlechteren Rückschluss auf die Zustandsänderung zulässt als z.B. der Crest-Faktor (Scheitelfaktor, Rangposition 4). Des Weiteren wird auch untersucht, in wie weit die Anzahl der verwendeten Parameter  $P$  Einfluss auf die Zustandserkennung hat. Bei Verwendung von nur 3 Parametern werden im Mittel 65% und bei 13 Parametern ca. 80% Erkennungsrate für die beiden Zustände erreicht. Bei steigender Anzahl der verwendeten Parameter erhöht sich die Erkennungsrate, auf ca. 80-85% bei 17 Parametern. Es folgt die Durchführung von zwei Messfahrten, das Fahrzeuginnenraumgeräusch wird jeweils aufgezeichnet. Die erste Messung wurde für den Zustand „auffälliges Geräusch“ durchgeführt, welches in der Datenbank bekannt ist (Klappern, Schlagen o.ä.). Nach der Messung wurden die auffälligen Bereiche markiert (ca. 25 Sekunden) und die Datensätze mit den Parame-

tern dafür berechnet (ca. 100 Datensätze). Die zweite Innenraumaufnahme wurde für den Zustand „normales Geräusch“ durchgeführt. Die Berechnung der Parameterdatensätze erfolgte analog zur ersten Messung für ca. 25 Sekunden. Die Berechnung der Parameter-Korrelationskennlinien, welche im Zusammenhang mit den spezifischen Korrelationskennlinien der Datenbank die Grundlage für die MSE-Ermittlung darstellen, erfolgt in Abhängigkeit der Faktoren  $D$  und  $\Delta D$ . Die Anzahl der Datensätze, die für die Berechnung der Korrelationskennlinien verwendet werden soll, wird durch den Faktor  $D$  beschrieben. Diese Datensätze werden stochastisch aus der Anzahl der  $\Delta D$  zur Verfügung stehenden Datensätze ausgewählt. Durch Variation der zwei Faktoren können Rückschlüsse über die Zuverlässigkeit der Zustandserkennung in Abhängigkeit der zur Verfügung stehenden Signallänge getroffen werden. Stehen weniger als 20 Datensätze (von max. 100) zur Berechnung der Parameterkorrelationskennlinien zur Verfügung, sinkt die Erkennungsrate unter 80%.

### 6.1.5 Auswertung der Korrelationsmethode

Anhand der vorherigen Analysen wurde eine Trennbarkeit von zwei Zuständen exemplarisch am Beispiel einer Mikrofonaufnahme durchgeführt. Es wurde nachgewiesen, dass sich die Korrelation reproduzierbar zwischen den Parametern ändert, wenn eine Zustandsänderung vorliegt. Die vorgestellte Methode (bzw. auch die verwendeten Merkmale) erlaubt somit die Trennung der definierten akustischen Zustände. In Tabelle 6.2 sind die mittleren Erkennungsraten für 2 auffällige und ein normales Innenraumgeräusch aus [TOB09] gezeigt. Die auffälligen Geräusche können dabei verschiedene Ursachen haben. Die normalen Geräusche sind hauptsächlich von den Umgebungsbedingungen geprägt.

Geräuschtyp	mittlere Erkennungsrate (%)
Auffällig: „schlagen“ (Zustand 1)	90,36
Auffällig: „wummern“ (Zustand 1)	77,67
Normal (Zustand 2)	84,84

Tabelle 6.2: Mittlere Erkennungsraten aus [TOB09]

Im Weiteren sollen die Erkenntnisse der hier vorgestellten Methode auf die kontinuierliche Signalanalyse mit selbstorganisierenden Karten erweitert und dadurch der Einfluss von Umgebungsgeräuschen minimiert werden.

## 6.2 Überwachte akustische Diagnose mit Self-Organizing-Maps

### 6.2.1 Theorie und Hintergrund

Der Einsatz von Kohonen-Netzen, auch als selbstorganisierende Karten bekannt (nach Kohonen, [Koh01]), wird für die Zuordnung diverser unbekannter akustischer Zustände verwendet. Die Zuordnung soll in drei Gruppen erfolgen:

- unbekanntes Geräusch
- fehlerbehaftetes Geräusch (Klasse 1 bis 11)
- fehlerfreies Geräusch (Klasse 12)

Die Kohonen-Karte besteht, wie in Abschnitt 4.4.4 erläutert, aus Neuronen, welche sich in ihren Gewichten in Abhängigkeit von den Eingabewerten ausrichten. Über die Verteilung der Gewichte im Raum können dann Klassenzugehörigkeiten abgebildet werden. Dieser Vorgang wird auch als Markierung oder Kennzeichnung (*Labeling*) bezeichnet. Über die Zustandsinformation des Trainingsdatensatzes werden die Neuronen markiert. Wenn nun ein unbekannter Datensatz verarbeitet wird, dann kann dieser über die Markierung des Neurons zugeordnet werden.

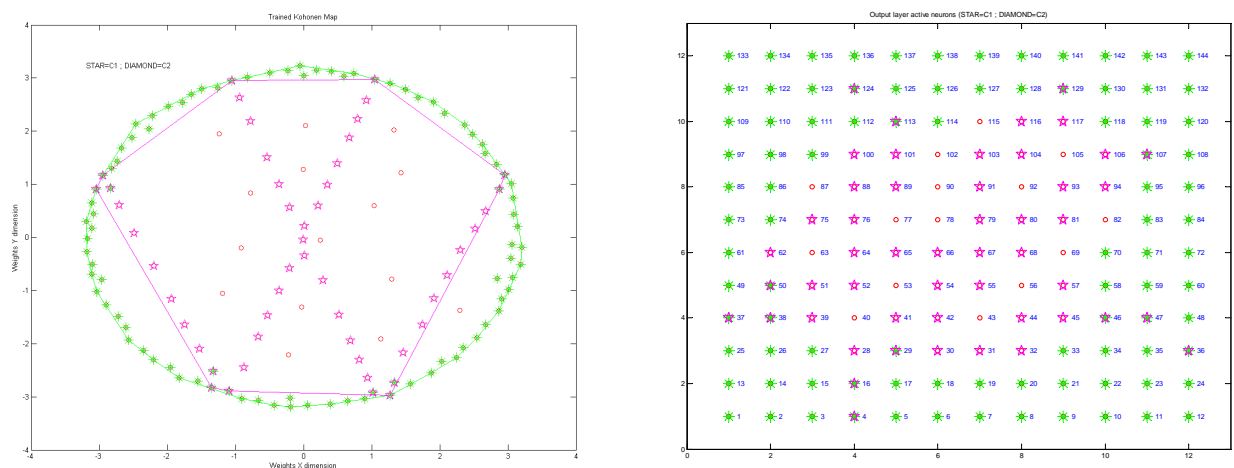


Abbildung 6.3: Beispiel SOM-Adaption:

links: Gewichte der Neuronen (zwei Klassen: C1 (Sternchen) und C2 (Diamanten))

rechts: Zweidimensionale Neuronenkarte mit Klassenzuordnung (markierte Neuronen)

In Abbildung 6.3a wurden die Neuronen, welche bei Anregung mit den Testdaten für Klasse C1 aktiv sind, mit einem grünem Stern dargestellt. Bei Anregung mit Testdaten der Klasse C2 wurden die aktivierten Neuronen mit einem magentafarbenem Diamanten gekennzeichnet. Dargestellt sind die Gewichte der Neuronen, welche zum Adaptionseende so angeordnet sind, dass Sie den Eingabedatenraum nachbilden. Dieses Netz hat zwei Eingabeneuronen und 12x12 Ausgabeneuronen in einer zweidimensionalen Netzstruktur. Die Karte des trainierten und markierten Netzwerkes ist in Abbildung 6.3b dargestellt. Dabei fallen zwei wichtige Punkte auf:

- nicht alle Neurone erhalten eine Klasseninformation durch den Markierungsprozess,

- nicht jedes Neuron repräsentiert unbedingt nur eine Klasse, auch Mehrfachzugehörigkeiten treten auf.

Um eine Klassifikation anhand der Kohonen-Karte durchführen zu können, interessieren die Neuronen, welche nur durch eine Klasse markiert sind; durch diese können unbekannte Eingabedaten eindeutig zugeordnet werden. Sollte keines dieser eindeutigen Neuronen ansprechen, so wird der Eingabedatensatz als unbekannt markiert. Des Weiteren können auch die eingangsabhängigen Neuronengewichte in der Komponentendarstellung (*component planes*, Komponentenschichten) ausgewertet werden, siehe Abbildung 6.2.1, wo eine qualitative Darstellung der Neuronengewichte in Abhängigkeit der Eingabemerkmale erfolgt. Bei Verwendung einer zweidimensionalen Ausrichtung der Neuronen bildet jedes Merkmal eine Komponentenschicht. Bei einer dreidimensionalen Ausrichtung hat jedes Merkmal entsprechend  $n_z$  Komponentenschichten, wobei  $n_z$  die Anzahl der Neuronen in z-Richtung ist.

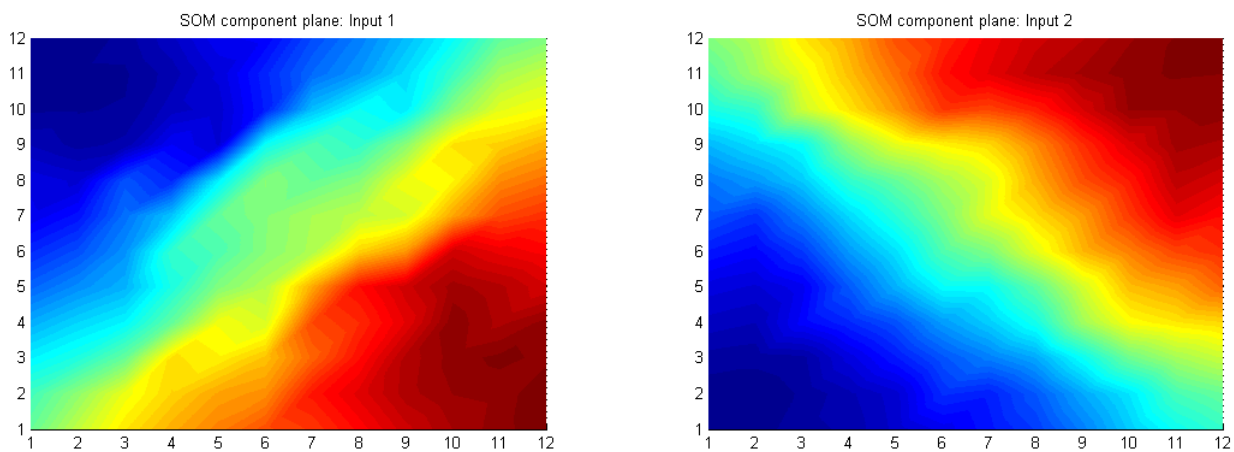


Abbildung 6.4: Komponentendarstellung (rot entspricht einer starken und blau einer geringen Ausprägung)

Eine weitere Darstellungsform ist die drei dimensionale Kohonen-Karte. Durch diese Art der Darstellung wird eine in Abhängigkeit der verwendeten Neuronen bestehende Anzahl von Kohonen-Karten gestapelt und somit ein 3D-Raum aufgebaut. Dadurch kann die Abbildung der Eingabewerte durch die Neuronen in einer zusätzlichen Dimension ausgewertet werden, was bei komplexen Klassifikationsproblemen eine bessere Klassenzuordnung ermöglicht. Der Zusammenhang zwischen Netztopologie und Erkennungsrate wird unter anderem im Abschnitt 6.2.3 untersucht.

### 6.2.2 SOM zur akustischen Zeitsignalbewertung und Fehlerdiagnose

Die bis hierher beschriebenen Methoden sollen nun auf das Problem der Zeitsignalklassifikation angewendet werden. Ziel ist ein akustischer SOM-Fehleragent, also die akustische Fehlererkennung basierend auf einer Kohonen-Karte. Dafür werden für die akustischen Signale fortwährend die psychoakustischen Merkmale berechnet. Diese ergeben ebenfalls zeitlich veränderliche Datenreihen, welche die Eingabeparameter der selbstorganisierende Karte sind. Durch Bestimmung der (in Abhängigkeit des Zustands) aktiven Neuronen können die akustische Zeitsignale bewertet und mit einer Zustandsinformation versehen werden. Für eine zuverlässige Zustandserkennung

werden periodisch die letzten 500 ms eines *Buffers* ausgewertet. Dazu werden stochastisch verteilt für ca. ein fünftel der zwischengespeicherten Daten die aktivierten Neuronen bestimmt und die Zustandsinformation ausgewertet. Als Rückgabewert des Modells wird folgendes erwartet:

- SOM-Ausgabewert  $s \in \mathbb{N}$  mit dem Wertebereich  $s = [0, 1, 2]$ , wobei
  - $s = 1$  dem fehlerbehafteten (im Folgenden als Klasse *C1*) und
  - $s = 2$  dem fehlerfreien (im Folgenden als Klasse *C2*) Zustand entspricht.
- SOM-Ausgabewert  $s = 0$ , wenn die Eingabeparameter kein Neuron angeregt haben. Null entspricht somit einem unbekannten Zustand (im Folgenden als Klasse *U*).

### Algorithmus zur Generierung des SOM-Ausgabewertes

In Kapitel 4.4.4 sind die Grundlagen zum Algorithmus der selbstorganisierenden Karte gegeben. An dieser Stelle wird also nicht auf die Ausbildung der Karte, sondern auf die Bewertung der Zeitsignaldaten durch die bereits trainierte Kohonen-Karte eingegangen. Aus dem Trainingsprozess liegen die Netzstruktur und Neuronengewichte vor, und diese werden durch den SOM-Agenten-Algorithmus 6.3 ausgewertet. Der dargestellte Pseudo-Code gilt für eine definierte Anzahl von Mikrofon-Abtastwerten. Für diese werden die Merkmale berechnet und in der Variable  $x$  zur Verfügung gestellt. Die Variable *SOM\_STEP* definiert die Länge eines Buffers und gleichzeitig die Häufigkeit der Ausführung der modulo-if-Schleife. Die Funktion *randperm* erzeugt zufällig verteilte Indexwerte von 1 bis *SOM\_STEP*. Die Funktion *ringbuffer* legt einen Ringspeicher an, ebenfalls mit der Länge von *SOM\_STEP*. In der *for*-Schleife wird der zeitliche Verlauf durch  $i$  bestimmt, und in jedem Takt wird der aktuelle Merkmalvektor in den Ringspeicher übertragen (Zeile 5). Wenn die modulo-Funktion 1 ist, bedeutet dies, dass der Ringspeicher gefüllt ist. Aus den im Ringspeicher befindlichen Merkmalsätzen wird anschließend eine zufällige Teilmenge (durch *rb\_idx*) der Größe  $T$  extrahiert (Zeile 7) und in *BUFFER* übertragen. Diese Merkmalsätze werden anschließend durch die SOM bewertet, weshalb die skalierten Merkmale (*scale*) an die Funktion *somtest* übergeben werden, welche in Algorithmus 6.4 erläutert wird. Als Rückgabewert gibt diese den Entscheidungswert des Abgleichs der Kohonen-Karte mit dem Merkmalvektor zurück. Dieser wird nun kontinuierlich in den SOM-Ausgabewert  $s$  und in geglätteter Form in  $s_{tp}$  geschrieben (Zeile 10-11, Gleichung 6.3).

---

### Algorithmus 6.3 SOM-Agent zur Bewertung von Mikrofonsignalen aus dem Fahrzeuginnenraum

---

*Eingabe:*

x: Merkmale der Mikrophon-Zeitsignalen des Mikrofons

SOM\_STEP: Schrittweite der Berechnung

T: Puffergröße

F: Glättungsfensterlänge

*Ausgabe:*

s: SOM-Ausgabewert

s\_tp: SOM-Ausgabewert geglättet

```
1  rb_idx=randperm(SOM_STEP);
2  rb=ringbuffer(SOM_STEP);
3  helper=NULL;
4  for i=1 to length(x) do
5      rb(new)=x;
6      if modulo(i,SOM_STEP) then
7          BUFFER=rb(rb_idx(1:T));
8          helper=somtest(scale(BUFFER));
9      end
10     s(i)=helper;
11     s_tp(i)=(1-1/F)*s_tp(i-1)+1/F*helper;
12 end
```

---

Die Funktion *somtest* erhält als Eingabewerte aus dem Trainingsprozess die Gewichte der Neuronen und die Klasseninformation. Alle übergebenen Merkmalsätze in Variable *data* werden in der for-Schleife analysiert. Dazu wird das Gewinnerneuron *wk* bestimmt, welches den geringsten Abstand zu dem aktuellen Merkmalsatz *data(i)* aufweist (Zeile 2). Die Markierung der Neuronen mit Klasse *C1*, *C2* liegt in Variable *mark* vor. In Zeile 3-7 wird verglichen, ob und welche Markierung das Gewinnerneuron aufweist. Wenn *wk* eine Markierung hat, dann wird dies in der Liste *label\_list* eingetragen. Sollte am Ende der for-Schleife kein Gewinnerneuron markiert sein, so wird der Wert 0 zurückgegeben, was dem Zustand "Unbekannt *U*" entspricht. Wenn Einträge in *label\_list* vorhanden sind, so wird der Mittelwert gebildet und als Ausgabewert zurückgegeben.



**Algorithmus 6.4** *somtest*-Funktion in 2 möglichen Realisierungsvarianten*Eingabe:*

w: Gewichte der Neuronen

mark: Klasseninfo der Neuronen

data: Merkmalsätze

*Ausgabe:*

out: Ausgabewert

```

1  for i=1 to length(data) do
2      wk=min(dist(data(i),w));
3      if mark(wk)==C1
4          label_list(i)=1;
5      else if mark(wk)==C2
6          label_list(i)=2;
7      end
8  end
9  if label_list == empty
10     out=0;
11 else
12     // Variante 1:
13     out=sum(label_list)/length(label_list);
14
15     // Variante 2:
16     if(length(find(label_list==1))>length(find(label_list==2)))
17         c=1;
18     else
19         c=2;
20     end
21 end

```

**Statistische und grafische Auswertemöglichkeiten**

Betrachtet man den Verlauf von  $s$  über der Zeit, so springt dieser bei *Variante 2* (siehe Algorithmus 6.4) zwischen 0, 1 und 2. Über die Häufigkeitsverteilung kann anschließend eine prozentuale Einteilung des Zeitsignals in die definierten Kategorien  $C1$ ,  $C2$  und  $U$  erfolgen. Für eine grafische Auswertung ist die diskrete Darstellung der Flankenwechsel ungeeignet. Zur Visualisierung wird deshalb eine tiefpassgefilterte Kennlinie  $s_{TP} \in \mathbb{R}$  von  $s$  berechnet. Durch die Tiefpassfilterung würde z.B. bei häufigen Flankenwechsel eine Stagnation der Kennlinie eintreten.

$$s_{TP}(n) = \left(1 - \frac{1}{L}\right) s_{TP}(n-1) + \frac{1}{L} s(n) \quad (6.3)$$

Eine statistische Auswertung der geglätteten Kennlinie ist auch denkbar, jedoch müssen dann Schwellenwerte definiert werden, die den unscharfen Bereich um die Klassen  $C1$ ,  $C2$  und  $U$  definieren. Für *Variante 1* ist diese Art der Auswertung ebenfalls notwendig. Die Berechnung der einzelnen Klassen-Auftrittswahrscheinlichkeiten erfolgt zum Beispiel anhand folgender Gleichungen.

$$\{C1, C2, U\} = \frac{1}{N} \cdot \sum_{i=1}^N check(s_{TP}(i), delta) \quad (6.4)$$

Bei Berechnung der Auftrittswahrscheinlichkeit des fehlerbehafteten Zustandes *C1* mit:

$$check_{C1}(s_{TP}(i), deltaC1) = \begin{cases} 1 & ; (1 - deltaC1) \leq s_{TP}(i) \leq (1 + deltaC1) \\ 0 & ; sonst \end{cases}$$

Bei Berechnung der Auftrittswahrscheinlichkeit des fehlerfreien Zustandes *C2* mit:

$$check_{C2}(s_{TP}(i), deltaC2) = \begin{cases} 1 & ; (2 - deltaC2) \leq s_{TP}(i) \\ 0 & ; sonst \end{cases}$$

Bei Berechnung der Auftrittswahrscheinlichkeit des unbekannten Zustandes *U* mit:

$$check_U(s_{TP}(i), deltaU) = \begin{cases} 1 & ; s_{TP}(i) \leq deltaU \\ 0 & ; sonst \end{cases}$$

Beträgt die Länge der akustischen Daten zum Beispiel 10 Sekunden, so werden über diese Zeit beispielsweise 20 Bewertungen bzw. Zustandszuordnungen vorgenommen. In Abbildung 6.5 ist der tiefpassgefilterte/ geglättete Verlauf des SOM-Ausgabewertes  $s_{TP}$  blau dargestellt. Dieses Beispiel wurde für ein fehlerbehaftetes Zeitsignal erstellt. Die grüne durchgezogene Linie markiert Bereiche, in denen ein fehlerbehaftetes Geräusch eindeutig hörbar ist. Die gestrichelten Linien stellen Schwellenwerte dar, welche die Bereiche der einzelnen Zustände definieren und für eine Berechnung der Auftrittswahrscheinlichkeiten (Gleichung 6.4) der Kategorien (grün=fehlerfrei; rot=fehlerbehaftet; gelb=unbekannt) benötigt werden. Für den Zustand 1 (entspricht *C1*), was einer Deklaration als fehlerbehaftetes Geräusch entspricht, beträgt die Auftrittswahrscheinlichkeit 0.7. Dies bedeutet, dass 70 Prozent des dargestellten Signalverlaufs dem Schwellenbereich des Zustandes 1 durch den SOM-Ausgabewert zugeordnet wurden. Für den Zustand 2 (fehlerfreies Geräusch) beträgt die Wahrscheinlichkeit 4 Prozent, und 20 Prozent wurden als unbekannt eingestuft. Die Differenz der Summe zu 100 Prozent beträgt rund 6 Prozent, welche außerhalb der definierten Schwellenbereiche liegt.

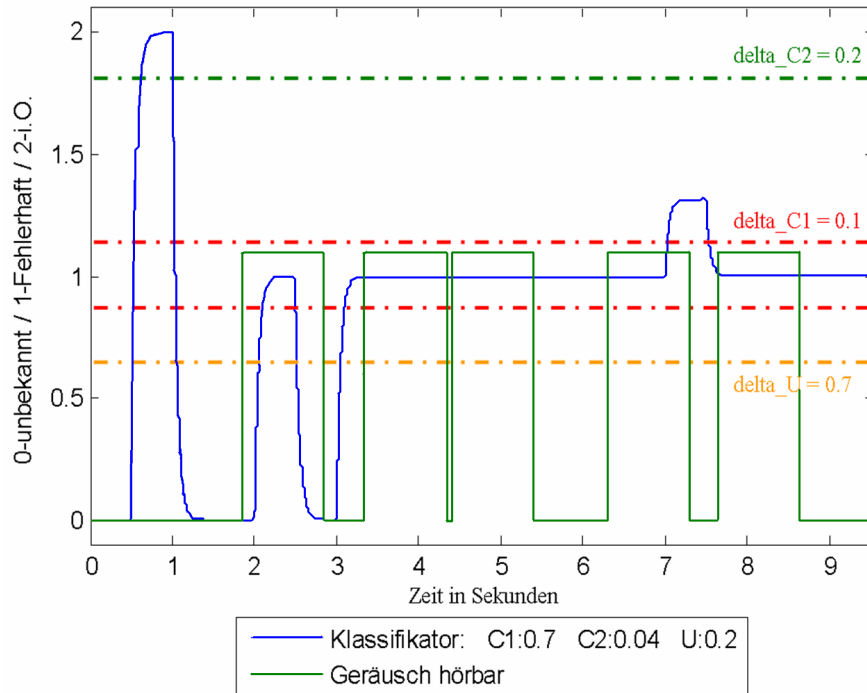


Abbildung 6.5: Bewertung eines Zeitsignals durch eine Kohonen-Karte

Das bewertete akustische Signal würde also eindeutig, mit einer Auftrittswahrscheinlichkeit von 70%, als fehlerbehaftetes Signal eingestuft werden. Tatsächlich handelt es sich auch um eine Innenraumaufnahme aus einem Fahrzeug mit einem defekten Radlager (Fehlerklasse 6, siehe Tabelle 5.1).

Man erkennt, dass für die Auswertung von  $s_{TP}$ , die Definition der Schwellenbereiche einen starken Einfluss auf die berechneten Auftrittswahrscheinlichkeiten hat. Ein optimaler Schwellenwertsatz könnte zum Beispiel für eine definierte Anzahl von Testsignalen evaluiert werden, indem diese iterativ so verändert werden, dass die Auftrittswahrscheinlichkeiten maximal werden. Alternativ können die Schwellenwerte  $\delta_{C1}$ ,  $\delta_{C2}$ ,  $\delta_U$  auch so modifiziert werden, dass eine stärkere Gewichtung einer Kategorie eintritt. Für folgende Untersuchungen wurde der ungefilterte SOM-Ausgabewert  $s$  für die Berechnung der Auftrittswahrscheinlichkeiten verwendet, um die Komplexität des Bewertungsvorgangs zu reduzieren. Die Berechnung erfolgt analog nach Gleichung 6.4, mit  $\delta = 0$  für alle Zustände.

Es werden folgende Parameter des SOM-Agenten variiert, um die Genauigkeit bei der Fehlerbewertung von akustischen Daten zu beurteilen:

- Dimension der Kohonen-Karte (2D, 3D)
- Anzahl der verwendeten Merkmale, die aus dem Zeitsignal berechnet und als Eingabewerte durch den Agenten beurteilt werden

Als Basis zum Trainieren des SOM-Fehleragenten dient die Geräuschdatenbank (siehe Abschnitt 5.1), welche aus berechneten Merkmalen für gut hörbare Sequenzen (in Abbildung 6.5 sind diese Bereiche zeitlich durch die grüne durchgezogene Linie markiert) in den akustischen Messdaten besteht. Diese wird in der Form abgeändert, dass die Fehlerklassen 1 bis 11 (siehe Tabelle

5.1) dem Zustand 1 zugeordnet werden. Die Fehlerklasse 12, welche dem fehlerfreien Zustand entspricht, wird dem Agenten als Zustand 2 präsentiert. Die Zeitkonstante bzw. Fensterlänge, für die die Merkmale jeweils berechnet werden, beträgt  $250ms$ . Zur Dimensionsreduktion bzw. Ranglistenstellung für die Merkmale wurde der *Relief*-Algorithmus (siehe Abschnitt 4.3.5 und Tabelle 6.3) verwendet. Das erste Merkmal in der Tabelle hat die höchste Priorität.

Roughness (Zeitverlauf)
Geschwindigkeit
Sharpness (Zeitverlauf)
Spectral Centroid
Giergeschwindigkeit
Loudness (Zeitverlauf)

Tabelle 6.3: Die 6 wichtigsten Merkmale aus dem *SEL* Datensatz, berechnet mit dem *Relief*-Algorithmus

Der Trainingsprozess läuft über  $T$  Trainingsdurchläufe, in welchen dem Netz ein zufällig ausgewählter Datensatz präsentiert wird und ist dann beendet, wenn die Gewichte der Neuronenkarte sich dem Stillstand annähern, also nicht weiter adaptieren.

Die originalen und vollständigen akustischen Messdaten dienen anschließend als Grundlage für die Analyse der Fehlerbewertung durch den Agenten. Dabei werden dem SOM-Agenten alle 129 akustischen Signale nacheinander präsentiert und die Auftrittswahrscheinlichkeiten der drei Zustände (in Abbildung 6.5 durch  $C1$ ,  $C2$  und  $U$  gekennzeichnet) berechnet.

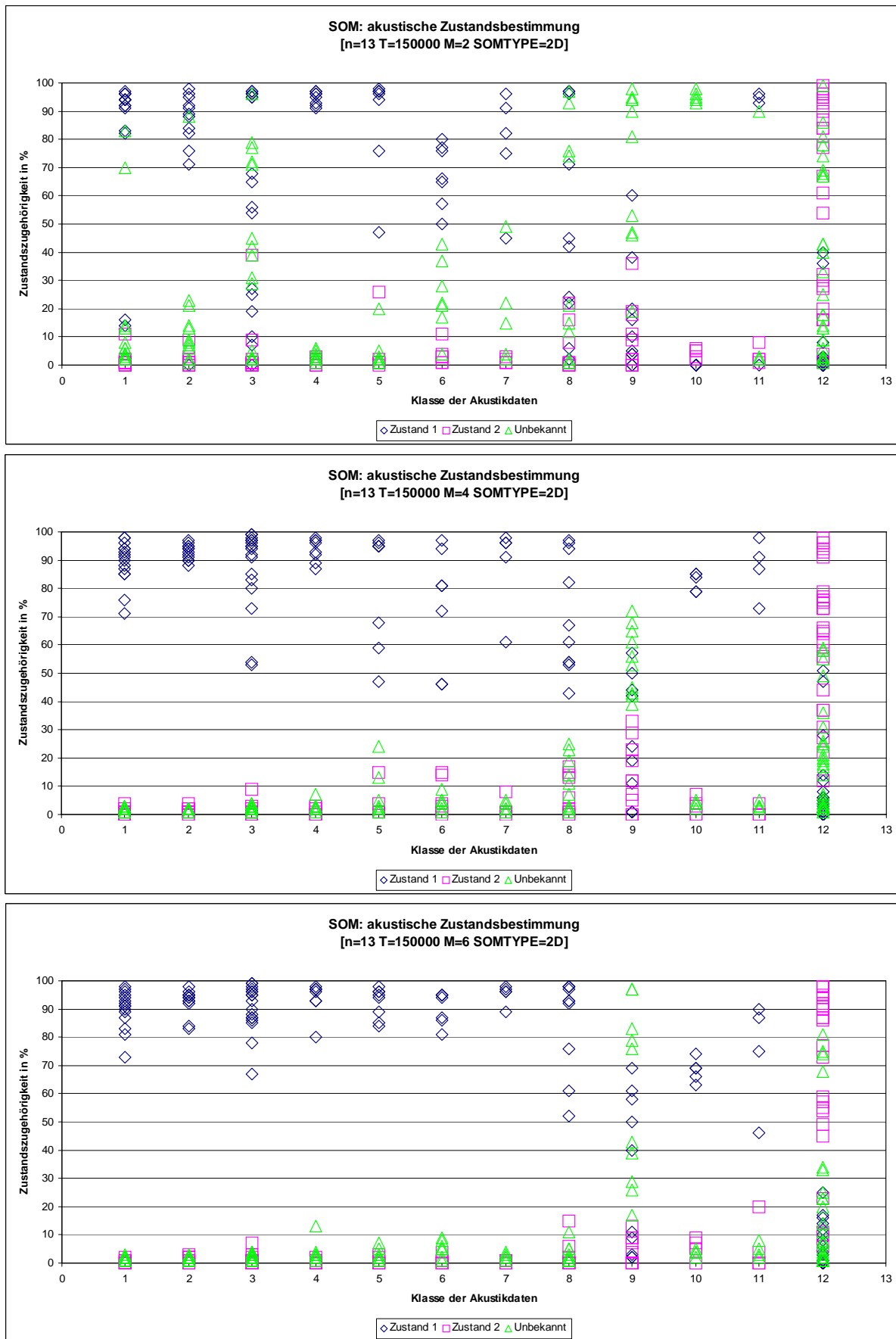
### 6.2.3 Test und Funktionsnachweis

In der ersten Phase wird der 2D-SOM-Agent getestet und die Anzahl der verwendeten Merkmale variiert. Die Anzahl der Neuronen wurde so gewählt, dass in dem Trainings- und Kennzeichnungsprozess möglichst viele markiert werden, jedoch nur wenige unmarkierte Neuronen in der Karte vorhanden sind. Für die 2D-Karte wurden  $n = 13$  Neuronen gewählt, normalverteilt initialisiert und mit 2, 4 und 6 Merkmalen aus dem *SEL* Datensatz (siehe Abschnitt 5.1) trainiert und getestet. Der Merkmalraum wird mit  $n^2$  Neuronen abgebildet. Die Ergebnisse in Tabelle 6.4 zeigen die Erkennungsraten für fehlerfreie und fehlerbehaftete Innenraumgeräusche. Für die fehlerfreien akustischen Messdaten wurden bei Verwendung von nur 2 Merkmalen 60% als Zustand 2 (fehlerfrei,  $C2$ ) erkannt. Von den restlichen 40% des Signals wurden jedoch nur 3.6% als Zustand 1 (fehlerbehaftet,  $C1$ ) erkannt und 34.8% als unbekannt ( $U$ ).

	fehlerfreie Innenraumgeräusche			fehlerbehaftete Innenraumgeräusche		
2D-SOM-Fehleragent	C1	C2	U	C1	C2	U
2 Merkmale	3.6	59.2	34.8	63.9	3.6	29.9
4 Merkmale	7.2	64.5	18.3	79.3	3.6	8.7
6 Merkmale	4.3	71.7	19.3	83.1	1.7	8.3

Tabelle 6.4: Auftrittswahrscheinlichkeiten des 2D-SOM-Fehleragenten für die 3 Zustände ( $C1$ = fehlerbehaftet,  $C2$ = fehlerfrei,  $U$ = Unbekannt) bei Variation der verwendeten Merkmale

Dies zeigt deutlich, dass der SOM-Fehleragent bei nur 2 Merkmalen einen Großteil der akustischen Daten nicht eindeutig zuordnen kann und diese deswegen als unbekannt deklariert. Bei Verwendung von 6 Merkmalen werden deutlich mehr Signalausschnitte den Zuständen 1 und 2 zugeordnet, so dass nur noch 19.3% als *U* eingestuft werden und *C2* auf 71.7% steigt. Für die fehlerbehafteten Messdaten zeigt sich ein analoges Bild bei Variation der Merkmale. Bei nur 2 verwendeten werden 29.95% als *U* erkannt und 63.9 % als *C1*. Werden hingegen 6 verwendet, so wird *C1* auf 83.1% erhöht und *U* sinkt entsprechend auf 8.3%.


Abbildung 6.6: Berechnete Auftrittswahrscheinlichkeiten des 2D-SOM-Fehleragenten bei Variation von  $M$

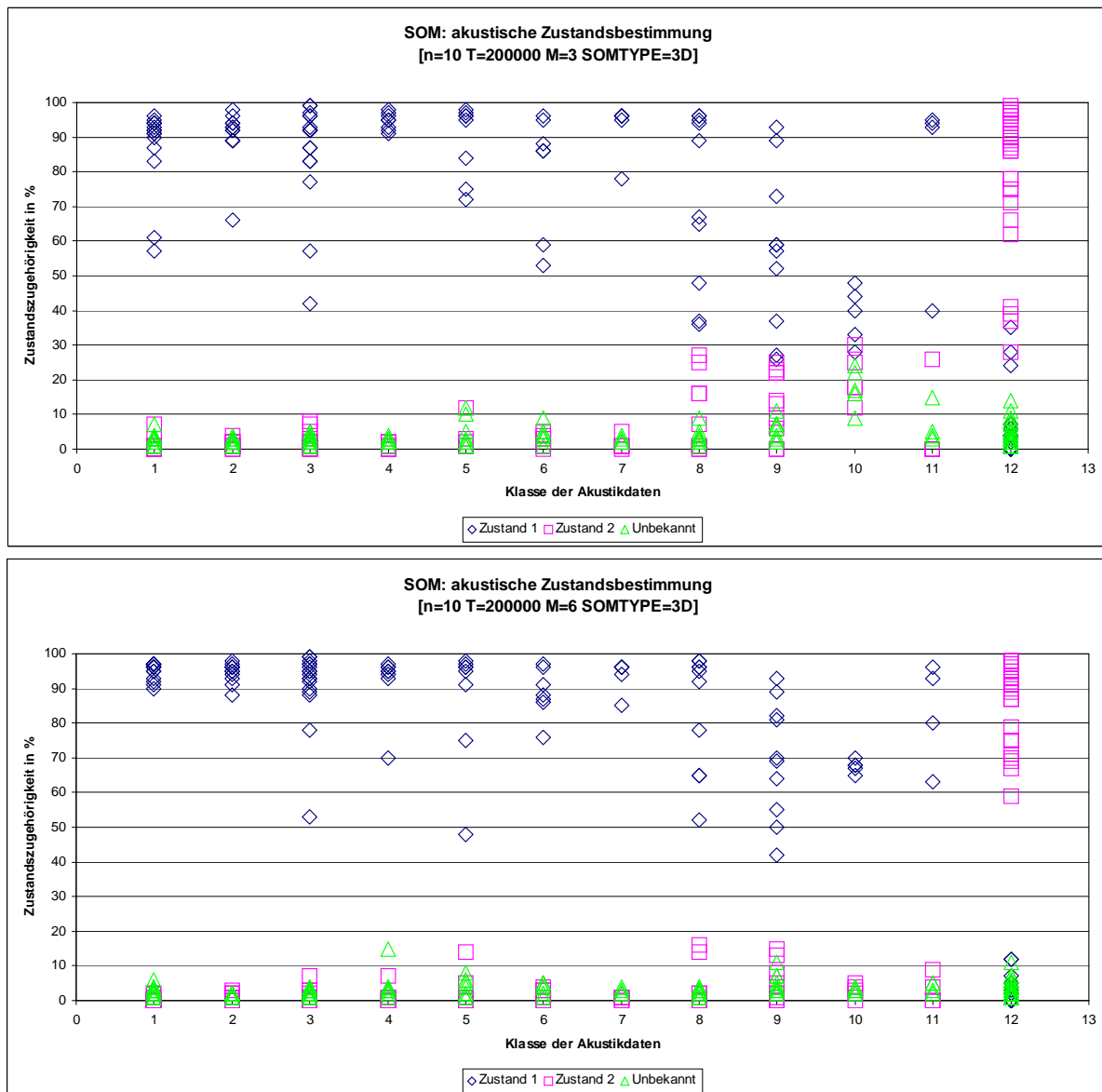
In Abbildung 6.6 sind die Auftrittswahrscheinlichkeiten für die 129 akustischen Signale dargestellt; dabei wurde jedes Signal wie in Abbildung 6.5 bewertet. Die Kohonen-Karte besteht aus 13 Neuronen und wurde über  $T = 150000$  Trainingsdurchläufe adaptiert. Es wird deutlich, dass mit steigender Anzahl der Merkmale  $M$  (von oben nach unten) die Zustandsentscheidungen genauer werden. Die Auftrittswahrscheinlichkeiten sind auf der y-Achse, die Klasse des Signals ist auf der x-Achse abgebildet. Die Klasse 1 bis 11 entsprechen dem Zustand 1, was einem fehlerbehaftetem akustischen Geräusch entspricht. Die Klasse 12 stellt den fehlerfreien akustischen Zustand 2 dar. Es ist deutlich erkennbar, dass die Erkennungsgenauigkeit des 2D-SOM-Fehleragenten mit steigender Anzahl an Merkmalen zunimmt, jedoch die Klassen 8 bis 11 nur schlecht dem richtigen Zustand zugeordnet werden. Der fehlerfreie Zustand 2, welcher durch die Klasse 12 repräsentiert wird, ist im Gegensatz zu den Klassen 1 bis 7 auch nur befriedigend genau erkannt. Viele Signale werden als unbekannt deklariert bzw. liegen in einem Bereich um 50%.

In der zweiten Phase folgt die Analyse und Vergleich des 3D-SOM-Fehleragenten. Durch diese erweiterte Dimension der Karte und die damit verbundene höhere Ausprägung der Gewichte zwischen den Neuronen und den Eingabewerten ist eine genauere Anpassung des Fehleragenten an den Merkmalraum zu erwarten. Die Auswertung erfolgt analog zum zweidimensionalen SOM-Fehleragenten. Das Training erfolgt auf den *SEL* Datensätzen, und zur Validierung werden die 129 akustischen Signalverläufe benutzt, aus denen die Datensätze extrahiert wurden (siehe Abschnitt 5.1).

	fehlerfreie Innenraumgeräusche			fehlerbehaftete Innenraumgeräusche		
3D-SOM-Fehleragent	C1 in %	C2 in %	U in %	C1 in %	C2 in %	U in %
3 Merkmale	4.8	79.6	4.2	80.8	4.7	4.1
6 Merkmale	1.8	87.3	3.4	87.4	1.9	2.9

Tabelle 6.5: Auftrittswahrscheinlichkeiten des 3D-SOM-Fehleragenten für die 3 Zustände (C1= fehlerbehaftet, C2= fehlerfrei, U= Unbekannt) bei Variation der verwendeten Merkmale

Die dreidimensionale Kohonen-Karte besteht aus  $n = 10$  Neuronen pro Dimension, in Summe wird der Merkmalraum also durch  $n^3$  Neuronen abgebildet. Bereits bei Verwendung von nur 3 Merkmalen wird eine deutlich bessere Erkennungsrate, von ca. 80% für beide Zustände, mit dem 3D-SOM-Fehleragenten gegenüber der 2D Version erreicht (Tabelle 6.4 und 6.5). Vor allem die fehlerfreien Innenraumgeräusche werden zuverlässiger erkannt, was sich in der Reduzierung der als unbekannt eingestuften Signalanteile auf unter 5% äußert. Steigt die Anzahl der Merkmale auf 6, so erhöht sich entsprechend die Zuordnungsrate für Zustand 1 (fehlerfreies Innenraumgeräusch) auf 87.3% und für den Zustand 2 (fehlerbehaftet) auf 87.4%. Die Fehlerkennungsrate (C2 bei tatsächlich C1 erkannt und umgekehrt) liegt unter 2%. In 6.7 sind die Ergebnisse detailliert (analog zu Abbildung 6.6) für den 3D-SOM-Agenten dargestellt. Die Kohonenkarte besteht aus  $10^3$  Neuronen und wurde über  $T = 200000$  Trainingsdurchläufe adaptiert. Es wird auch hier deutlich, dass mit steigender Anzahl der Merkmale  $M$  (von oben nach unten) die Zustandsentscheidungen genauer werden.


Abbildung 6.7: Berechnete Auftretswahrscheinlichkeiten des 3D-SOM-Fehleragenten bei Variation von  $M$ 

## 6.2.4 Modellparameter und Optimierungen

### a) Quantisierungsfehler in Abhängigkeit der Epochenzahl:

Der Einfluss der Trainingsepochen auf die Ausprägung der Neuronen in der Karte wird im Folgenden analysiert. Dazu wird eine Netztopologie gewählt (3D mit 10x10x10 Neuronen) und anschließend die Anzahl der Trainingsepochen variiert. Als Maß für die Güte der Ausprägung wird zum einen der Quantisierungsfehler (Gleichung 4.43) und zum anderen die Verteilung der Neuronen analysiert. Für letzteres findet eine Auswertung der absoluten Häufigkeit der Gewinnerneuronen für den Trainingsdatensatz statt. Für ein gut ausgeprägtes Netz wird erwartet, dass eine Gleichverteilung auftritt. Um diese Gleichverteilung zu bewerten, wird die Varianz als Maß für die Streuung verwendet.



$$\text{var}(\text{hist}) = \sigma_{\text{hist}}^2 = \frac{1}{N-1} \sum_{i=1}^N \left( \text{hist}_i - \overline{\text{hist}} \right)^2 \quad (6.5)$$

Die Parameter *Lernrate* und *Scheitelbreite* wurden wie folgt gewählt:

- Start-Schrittweite  $\mu_I = 0.9$  und End-Schrittweite  $\mu_F = 0.1$
- Start-Scheitelbreite  $\sigma_I = 3$  und End-Scheitelbreite  $\sigma_F = 0.1$

Diese entsprechen einer allgemeinen Initialisierung der Parameter, die für skalierte Datensätze (siehe Gleichung 5.1) gute Ergebnisse erreicht.

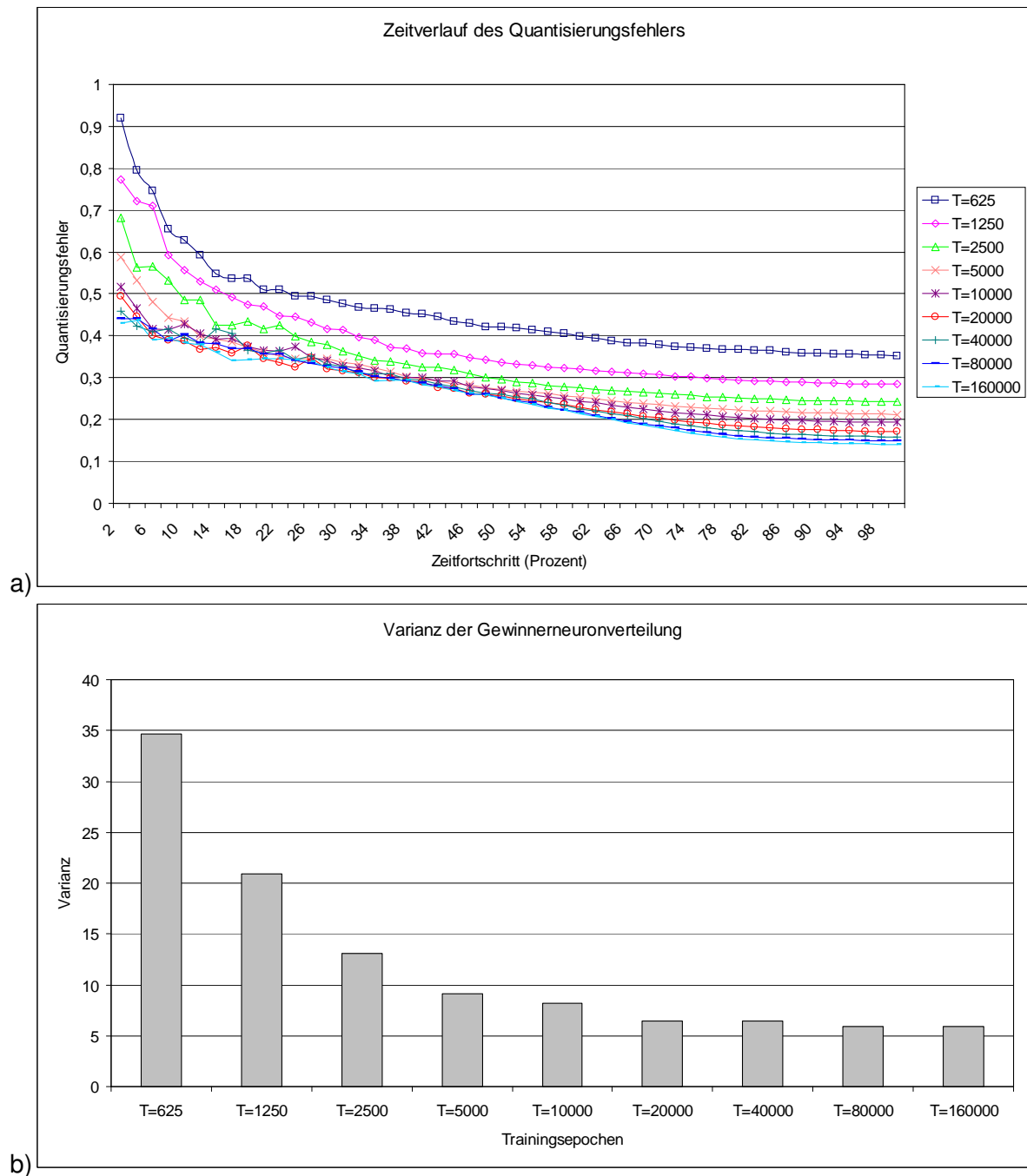


Abbildung 6.8: a) Quantisierungsfehler bei verschiedenen Epochenlängen des Trainingsprozesses  
b) Varianz aus der absoluten Häufigkeit der Gewinnerneuronen für den Trainingsdatensatz

Der Quantisierungsfehler gibt den mittleren Abstand der Gewinnerneuronen zu den Eingabewerten des Trainingsdatensatzes an und ist in Abbildung 6.8a gezeigt, wobei die Anzahl der Epochen  $T$  im Bereich 625 bis 160000 variiert wird. Es wird davon ausgegangen, dass ein geringer Quantisierungsfehler zum Adaptionseende - auf der y-Achse ist der Adaptionzeitverlauf in Prozent angegeben - eine gute Abbildung des Werteraums durch die Neuronen bestätigt. Es ist ersichtlich, dass eine Sättigung in der Minderung des Fehlers bei ca.  $T = 20000$  Epochen eintritt. Die Veränderung gegenüber einer höheren Epochenzahl ist dann nur noch von marginaler Bedeutung. Betrachtet man die Varianz in Abbildung 6.8b, so zeigt sich ebenfalls eine Sättigung ab ca. 20000 Epochen. Es ist erkennbar, dass die Varianz sehr groß ist bei einem schlecht ausgeprägten Neuronennetz. In Abbildung sind zwei Kohonenkarten unterschiedlicher Ausprägung durch ihr Histogramm miteinander verglichen. Werden häufig die gleichen Neuronen angeregt, so ist die Varianz sehr hoch. Bei einer - an den Trainingsdatensatz - gut angepassten Neuronenkarte ist die Varianz klein und die absolute Häufigkeit nähert sich einer Gleichverteilung an.

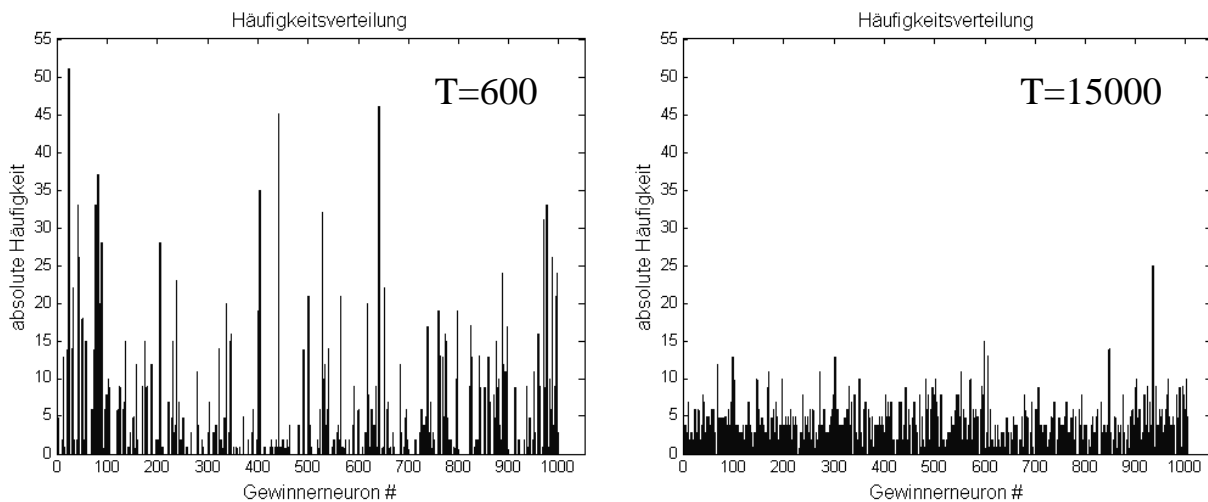


Abbildung 6.9: Das Histogramm für 600 Trainingsepochen zeigt im Vergleich zu 15000 Epochen eine deutlich schlechtere Verteilung. Dementsprechend ist die Varianz für  $T=600$  größer als für  $T=15000$ .

Es ist somit gezeigt, dass die Varianz als zusätzliches Maß zur Beurteilung der Ausprägung einer Neuronenkarte verwendet werden kann.

## b) Quantisierungsfehler in Abhängigkeit der Lernrate und Scheitelbreite

Ausgehend von den in Abschnitt 6.2.4a genannten Parametern wird ein optimierter Parametersatz ermittelt. Als Maß für die Güte des Parametersatzes wird der Quantisierungsfehler und die Varianz gewählt. Die Initialisierung der Neuronen-Gewichte erfolgt normalverteilt, mit einem Mittelwert von 0 und einer Standardabweichung von 1.

Die Optimierung bezieht sich auf die Analyse des Quantisierungsfehlers und iterativer Veränderung obiger Parameter. Die Untersuchungen werden bei gleichbleibender Neuronenzahl durchgeführt, da laut Definition der Quantisierungsfehler abnimmt, wenn die Neuronen steigen, da der Raum besser abgedeckt wird. Die Trainingszeit wird ebenfalls konstant gehalten bei  $T = 10000$ . Die

Scheitelbreite und die Schrittweite werden im Algorithmus linear vom Startwert auf den Stopwert verringert.

- Variation des Start-Scheitelfaktors  $\sigma_I$  mit  $\mu_I = 0.9$   $\mu_F = 0.1$   $\sigma_F = 0.1$  :

Es ist in Abbildung 6.10 ersichtlich, dass gerade für große Scheitelbreiten sehr starke Schwankungen des Quantisierungsfehlers am Anfang des Adaptionprozesses auftreten. Diese führen auch zu einem höheren Fehler am Trainingsende. Die Schwankungen sind auch im Verlauf der Varianz ( $\sigma_{hist}^2$ , in der Legende in Klammern) ersichtlich. Interessant ist, dass bei zu geringer Start-Scheitelbreite der Quantisierungsfehler zwar einigermaßen im Mittelfeld liegt, dennoch die Varianz deutlich größer ist (z.B. für  $\sigma_I = 0.5$  mit  $\sigma_{hist}^2 = 32.7$  ca. 3 mal so hoch wie der Durchschnitt). Für einen Wert von  $\sigma_I = 1$  wird der geringste Fehler bei kleinster Varianz erreicht.

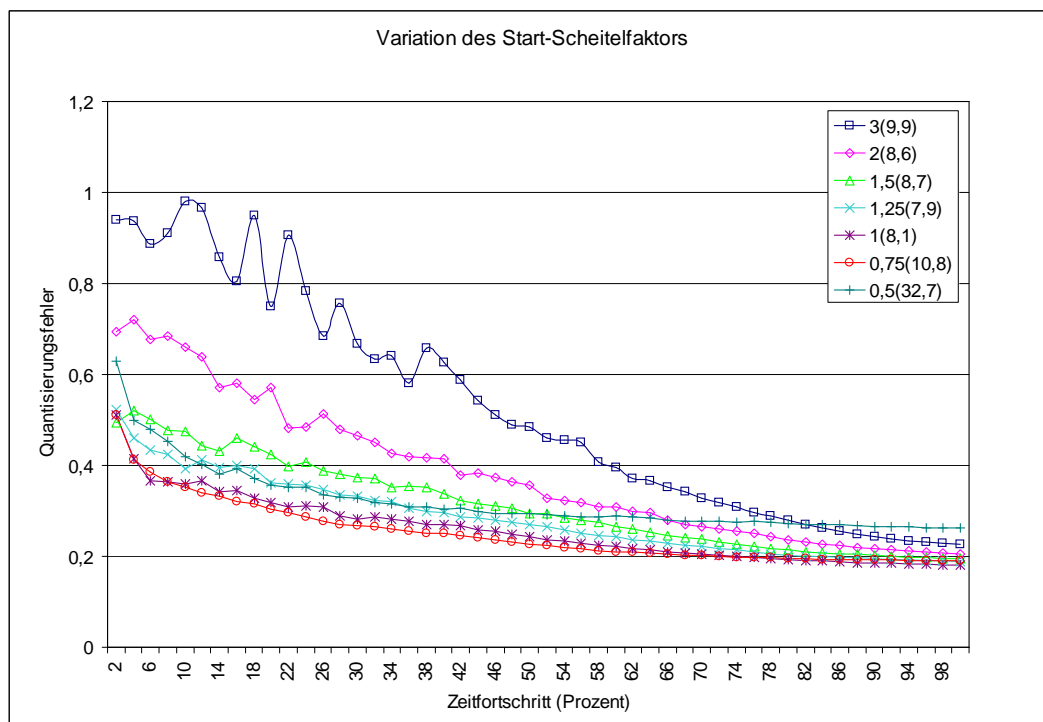


Abbildung 6.10: Variation von  $\sigma_I$  im Bereich  $[0,5...3]$

- Variation des End-Scheitelfaktors  $\sigma_F$  mit  $\mu_I = 0.9$   $\mu_F = 0.1$   $\sigma_I = 1$  :

Abbildung 6.11 zeigt deutliche Unterschiede im Verlauf des Quantisierungsfehlers und der Varianz ( $\sigma_{hist}^2$ , in der Legende in Klammern) am Ende der Adaption. Es wird nachgewiesen, dass ab einem End-Scheitelwert kleiner  $\sigma_F = 0.1$  kaum eine Veränderung der Varianz und des Quantisierungsfehlers auftritt. Für einen Wert von  $\sigma_F = 0.01$  wird der geringste Fehler bei kleinster Varianz erreicht.

- Variation der Start-Schrittweite  $\mu_I$  mit  $\sigma_I = 1$   $\sigma_F = 0.01$   $\mu_F = 0.1$  :

Aus den Kennlinien in Abbildung 6.12 geht hervor, dass die Start-Schrittweite möglichst groß gewählt werden sollte. Es zeigen sich wenige Unterschiede im Verlauf des Quantisierungsfehlers und der Varianz ( $\sigma_{hist}^2$ , in der Legende in Klammern). Für  $\mu_I = 1$  wird der

kleinste Quantisierungsfehler bei kleinster Varianz erreicht. Die Varianz und die Fluktuation des Quantisierungsfehlers (am Adaptionsbeginn) nimmt mit kleiner werdendem  $\mu_I$  zu.

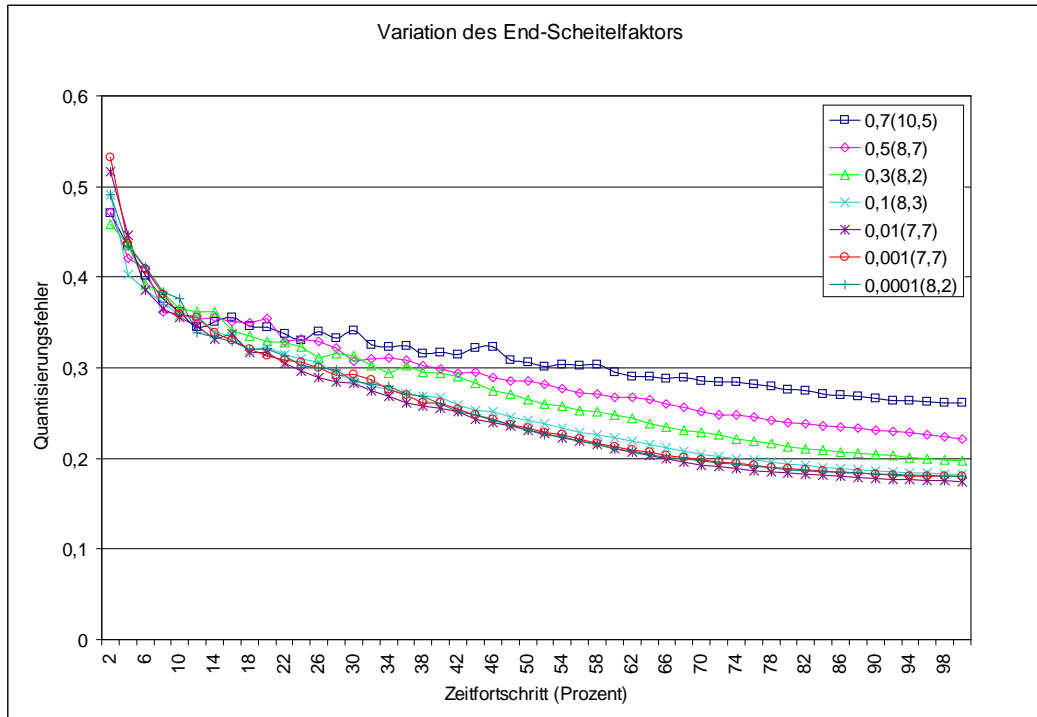


Abbildung 6.11: Variation von  $\sigma_F$  im Bereich [0.0001 ... 0.7]

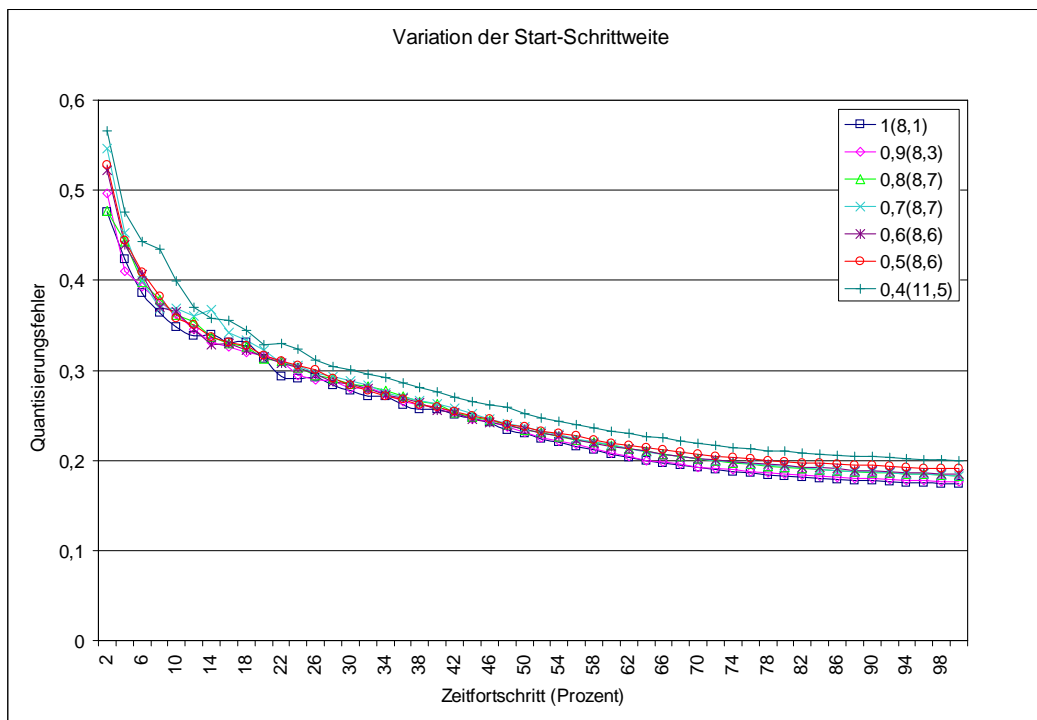


Abbildung 6.12: Variation von  $\mu_I$  im Bereich [0.4 ... 1]

- Variation der End-Schrittweite  $\mu_F$  mit  $\sigma_I = 1$   $\sigma_F = 0.01$   $\mu_I = 1$  :

Es zeigen sich in Abbildung 6.13 annähernd gleiche Verläufe des Quantisierungsfehlers und der Varianz ( $\sigma_{hist}^2$ , in der Legende in Klammern) für alle untersuchten  $\mu_F$ . Die Varianz befindet sich durchgängig auf gleichem Niveau, so dass keine Tendenzen abgeleitet werden können. Prinzipiell zeigt sich dadurch, dass die End-Schrittweite wenig Einfluss aufweist. Für einen Wert von  $\mu_F = 0.5$  wird der geringste Fehler bei kleinster Varianz erreicht.

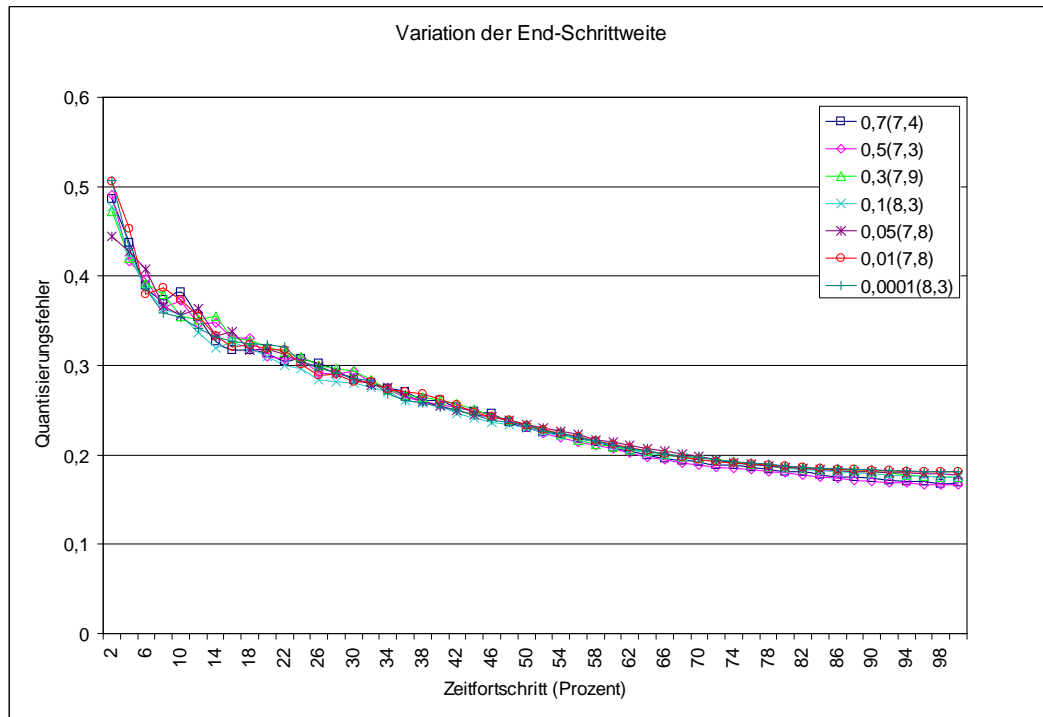


Abbildung 6.13: Variation von  $\mu_F$  im Bereich [0.0001 ... 0.7]

Zusammengefasst für alle Analysen wurden folgende Parameter als optimale Parameter definiert:

$$\mu_I = 1; \quad \mu_F = 0.1; \quad \sigma_I = 1; \quad \sigma_F = 0.01$$

Wie im Kapitel 4.4.4 erläutert und in diesem Abschnitt gezeigt wurde, hat die Scheitelbreite einen großen Einfluss auf den Adaptionvorgang. Wenn Parameter geändert werden, so ist vor allem eine erneute Anpassung der Start-Scheitelbreite notwendig, da diese direkt mit dem Wertebereich der Merkmale (Ausprägung des Eingaberaumes) zusammenhängt. Diese hat maßgebenden Einfluss auf die Ausprägung der Neuronen hinsichtlich gleichmäßiger Verteilung der Neuronen. Deshalb sollte die Scheitelbreite nicht zu klein gewählt werden (siehe auch [Neu98]).

Eine automatisierte Parametrisierung ist bei einer Veränderung des Parametersatzes wünschenswert. In [HG01] wurde ein Kalman-Filter zur rekursiven Bestimmung der Parameter eingesetzt und als *Auto-SOM* bezeichnet. Eine solche Methode rechtfertigt den zusätzlichen Rechenaufwand, wenn Datensätze automatisiert ausgewertet werden sollen und eine kurze Adaptionzeit gefordert wird. Eine einfachere verallgemeinerte Methode, welche zwar nicht die schnellstmögliche Adaptionszeit, aber eine Gleichverteilung der Neuronen bezweckt, ist wie folgt realisierbar. Für die Matrix  $X$ , welche die Musterdatensätze mit der Struktur [Muster, Merkmale] für den Trainingsprozess

repräsentiert, wird der Maximalwert ermittelt. Dieser dient als Richtwert für die Ausdehnung des Eingaberaumes. Für die in diesem Abschnitt durchgeführten Analysen für die optimierte Start-Scheitelbreite ergibt sich  $\sigma_I = 1 \approx 0.2 \cdot \max(X)$ . Der Algorithmus wird also dahingehend verändert, dass sich die Start-Scheitelbreite anhand einer fixen Relation zur maximalen Ausdehnung des Eingaberaumes ergibt.

### c) Einfluss der Puffergrößen

Wie im Algorithmus in Abschnitt 6.3 ersichtlich ist, bestimmen zwei Kenngrößen die Auswahl der Merkmalsätze, welche der SOM-Agent analysiert. Zum einen die Berechnungsschrittweite *SOM\_STEP* und zum anderen die Puffergröße *T*. Beide Größen beeinflussen die Berechnung/Auswahl der Merkmalsätze, welche durch den SOM-Agenten analysiert werden. Ihr Einfluss wird in dem Schema in Abbildung 6.14 ersichtlich. *SOM\_STEP* beeinflusst zum einen die Größe des Ringspeichers *rb* und damit auch die Länge des zufällig verteilten Indexvektors *rb\_idx*. Durch die Größe des Ringspeichers wird die Anzahl der durchgeführten SOM-Agent Analysen beeinflusst. Folglich wird zum Beispiel bei einem kleinen *SOM\_STEP* häufig ein neuer SOM-Agent-Ausgabewert generiert und damit auch mehr Rechenzeit benötigt. Die Puffergröße *T* gibt vor, wie viele Merkmalsätze aus dem Ringspeicher zufällig verteilt extrahiert und damit zur Analyse bereitgestellt werden. Diese Größe beeinflusst damit auch direkt den Rechenaufwand; umso mehr Merkmalsätze analysiert werden sollen, desto mehr Rechenzeit wird benötigt.

Merkmal	Abtastrate in ms (gerundet)
Roughness (Zeitverlauf)	0.065
Geschwindigkeit	0.002
Sharpness (Zeitverlauf)	4.162
Spectral Centroid	6.404
Giergeschwindigkeit	0.002
Loudness (Zeitverlauf)	4.162

Tabelle 6.6: Die 6 wichtigsten *Relief*-Merkmale aus dem *SEL* Datensatz und die dazugehörigen Abtastraten

Die Merkmale liegen mit unterschiedlichen Abtastraten vor. Für folgende Analysen dienen die CAN-Daten mit 48 kHz als Referenz, alle anderen Merkmale werden sinnbildlich entsprechend dem *sample-and-hold* Prinzip neu abgetastet, um eine gleichwertige Zeitskala für alle Merkmale zu erreichen. Wie in Tabelle 6.6 hervorgeht, ist die Abtastung der CAN-Daten mit 48 kHz (ca. 0.002 ms) nicht sinnvoll. Vorwiegend werden im Bereich des Fahrzeug-Antriebs-CAN Zykluszeiten im Millisekundenbereich erwartet. Das verwendete Messsystem hat die CAN-Daten jedoch analog zu den Mikrofonkanälen mit 48 kHz abgetastet. Neben einem erhöhten Speicherbedarf ergibt sich daraus auch der beschriebene Mehraufwand zum Zeitskalenabgleich, welcher bei der Auslegung eines mobilen Messsystems berücksichtigt werden muss (siehe Kapitel 7). Für die folgenden Auswertungen wurde der 3D-SOM-Agent verwendet (10x10x10 Neuronen).

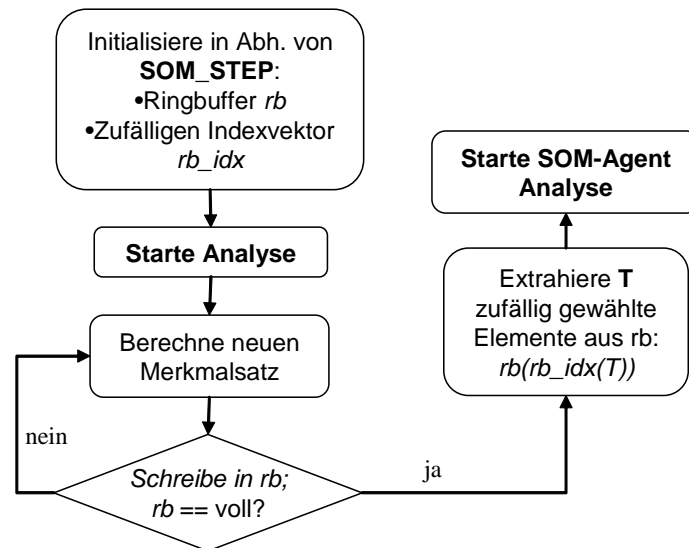


Abbildung 6.14: Schema zur Interpretation der der Berechnungsschrittweite  $SOM\_STEP$  und der Puffergröße  $T$

Im Folgenden werden beide Kenngrößen variiert, um die optimalen Werte für  $SOM\_STEP$  und  $T$  zu ermitteln und damit zum Beispiel den Rechenaufwand nicht unangemessen - durch eine zu kleine Schrittweite oder zu viele Merkmalsätze - zu erhöhen. In Abbildung 6.15 wird der Einfluss von  $T$  auf die Erkennungsraten bei einer festen Berechnungsschrittweite  $SOM\_STEP$  von 500 ms gezeigt. Die x-Achse stellt die jeweilige Puffergröße dar und die y-Achse die gemittelte Erkennungsrate für die in der Legende gekennzeichneten akustischen Zustände (außerhalb der Klammern sind die SOM-Agent Zustände und innerhalb die tatsächlichen Zustände der Messdaten gekennzeichnet). Es wird deutlich, dass bei  $T$  kleiner 5 ms die mittlere Erkennungsrate für "C1" sinkt. Ab  $T=0.25$  ms fällt auch die Rate für "C2". Zusätzlich ist ersichtlich, dass die Fehlinterpretation und Einstufung als Unbekannt bei einer zu kleinen Puffergröße  $T$  ansteigt.

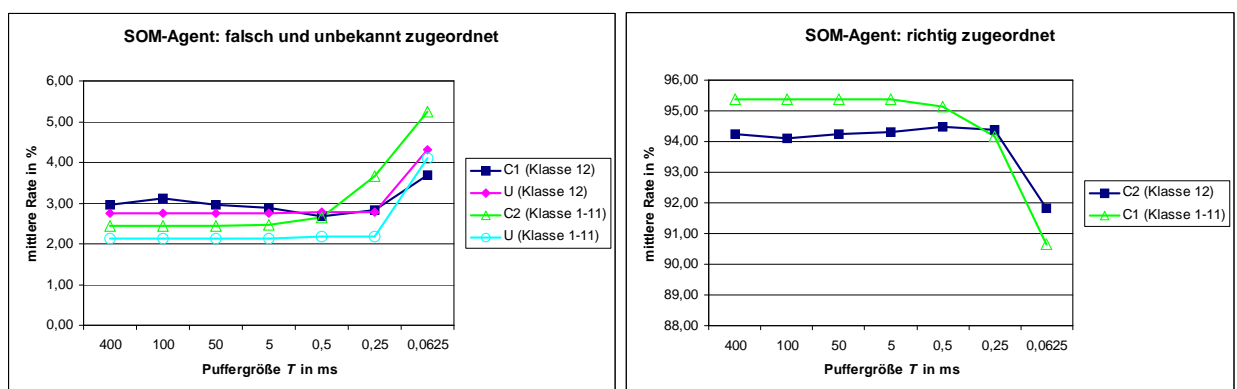


Abbildung 6.15: Analyse des Einflusses der Puffergröße von  $T$  auf die Erkennungsraten des SOM-Agents

Bei der Analyse der variierenden Berechnungsschrittweite  $SOM\_STEP$  in Abbildung 6.16 (bei  $T=5$  ms) geht hervor, dass bei  $SOM\_STEP$  größer 500 ms und kleiner 100 ms die Erkennungsraten abnehmen. Die x-Achse bildet die jeweilige Schrittweite ab und die y-Achse die gemittelte Erkennungsrate für die in der Legende gekennzeichneten akustischen Zustände. Auf der anderen Seite

ist in den Grafiken ersichtlich, dass bei abnehmender Zeit die Fehlinterpretation (Abbildung 6.16 links, “C1 [Klasse 1 ... 11]” und “C2 [Klasse 12]” ) ansteigt. Der Schnittpunkt beider Verläufe ergibt sich bei 500 ms.

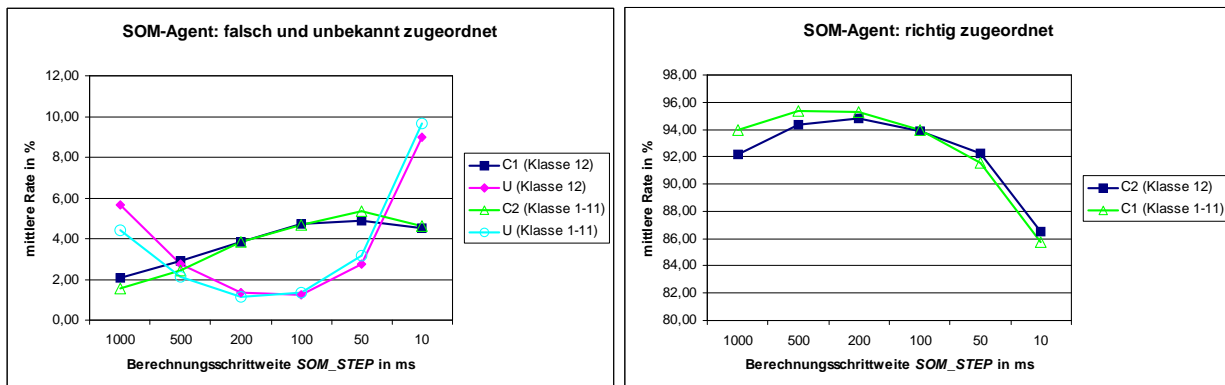


Abbildung 6.16: Auswertung der Erkennungsraten des SOM-Agents bei variierendem  $SOM\_STEP$

Zusammenfassend hat sich gezeigt, dass die Puffergröße  $T$  im Bereich 5 ms gewählt werden sollte. Die Berechnungsschrittweite erreicht bei 500 ms den besten Kompromiss aus Fehlinterpretierten und richtig zugeordneten Messdaten.

#### d) Analyse der “somtest”-Funktion

Die in Algorithmus 6.4 gezeigten Realisierungsvarianten *Variante 1* und *Variante 2* werden hinsichtlich ihrer Erkennungsraten analysiert. In Tabelle 6.7 sind die mittleren Entscheidungsraten des 3D-SOM-Fehleragenten für die verschiedenen Geräuscharten - welche aus 129 Mikrofonaufnahmen ausgewertet wurden - bei zwei verschiedenen Algorithmusvarianten (siehe Algorithmus 6.4) dargestellt. Es wird deutlich, dass *Variante 2* bessere Ergebnisse erreicht. Gerade bei einem in *Variante 1* schwierig zu beurteilenden Fehlergeräusch zeigt sich die Einfachheit der *Variante 2* im Vorteil, da eine unscharfe Bewertung (siehe Absatz der Gleichung 6.4) die Wahl optimierter Schwellenwerte voraussetzt. Dies zeigt auch, dass eine Parametrisierung der Schwellen bei *Variante 1* sehr schwierig ist. *Variante 2* hat somit den Vorteil, dass das Modell nicht komplexer wird und auf Anhieb sehr gute Erkennungsraten mit wenig Optimierungspotenzial erreicht werden. Sollten jedoch bei der Auswertung der zu analysierenden Datensätze Optimierungspotenziale ersichtlich sein, so kann über eine *Variante 1* mit angepassten Schwellenwerte durchaus eine Verbesserung der Erkennungsraten erreicht werden.



Fehlerart	Klasse	Variante 1 (%)	Variante 2 (%)
defektes Motorlager	1	92.7	98.5
Koppelstange ausgeschlagen rechts	2	94.9	98.4
Koppelstange ausgeschlagen links	3	87.6	97.4
Radlager stark beschädigt rechts	4	94.5	97.4
Radlager stark beschädigt links	5	84.5	95.1
defektes Radlager links	6	89.0	95.6
defektes Radlager rechts	7	92.6	97.6
Koppelstange lose geschraubt links	8	80.3	93.8
defekte Gelenkwelle Außenlager rechts	9	51.6	90.6
defektes Federbeinlager rechts	10	52.0	95.8
defektes Federbeinlager links	11	72.3	89.0
fehlerfrei/ Originalklang	12	89.2	94.1

Tabelle 6.7: Mittlere Entscheidungsraten des 3D-SOM-Fehleragenten für 2 Algorithmusvarianten

Es fällt auf, dass bei *Variante 1* die Häufigkeit der aktivierten Neuronen klassenabhängig auf den SOM-Ausgabewert abgebildet wird. Diese zusätzliche Information hat auf den für den SOM-Agenten gewünschten Ausgabewert keinen Nutzen und wird deswegen auch nicht empfohlen. Bei der Analyse der Ausbildung der Neuronenkarte kann diese Information jedoch nützlich sein und Aufschluss über die Ausprägung der Karte - in Abhängigkeit variierteter Parameter - liefern.

#### e) Einfluss der Netz-Topologie und Neuronenzahl auf die Ausprägung der Neuronen-Karte

In Kapitel 4.4.4 wurden die Topologiearten der SOM vorgestellt. Für die Realisierung des SOM-Agenten wurden eine zwei- und dreidimensionale Topologie gewählt. Die Neuronenanzahl berechnet sich über  $n^d$ , wobei  $n$  die Neuronen pro Dimension und  $d$  die Anzahl der Dimensionen beschreibt. So besteht ein 2D-SOM-Agent mit  $n = 10$  aus 100 Neuronen und ein 3D-SOM-Agent aus 1000 Neuronen. Abbildung 6.17 zeigt den Quantisierungsfehler über die Neuronenanzahl für einen 2D- und 3D-SOM-Agenten für unterschiedliche Adaptiongrade über zwei verschiedene Trainingsepochen  $T = 1000$  und  $T = 100000$ . Es wird deutlich, dass die Topologie bei diesem Vergleich keinen maßgeblichen Einfluss auf den Quantisierungsfehler hat.

Dennoch ist in Abbildung 6.18 ersichtlich, dass die 3D-Topologie vor allem bei einer geringen Trainingsepochenanzahl ( $T = 1000$ ) einen höheren Anteil markierter Neuronen aufweist, die Adaption dementsprechend gezielter erfolgt. Markierte Neuronen erhielten durch den Markierungs-Prozess (*labeling*, Abschnitt 4.4.4) die Information zur Klassenzugehörigkeit (fehlerfrei oder fehlerbehaftet). Wird die Trainingszeit jedoch auf 100000 erhöht, so weist die 2D-SOM einen höheren Anteil, von ca. 10% an markierten Neuronen auf. Dennoch ist im Vergleich zur vorigen Abbildung nachgewiesen, dass diese Differenz keinen spürbaren Einfluss auf den Quantisierungsfehler hat.

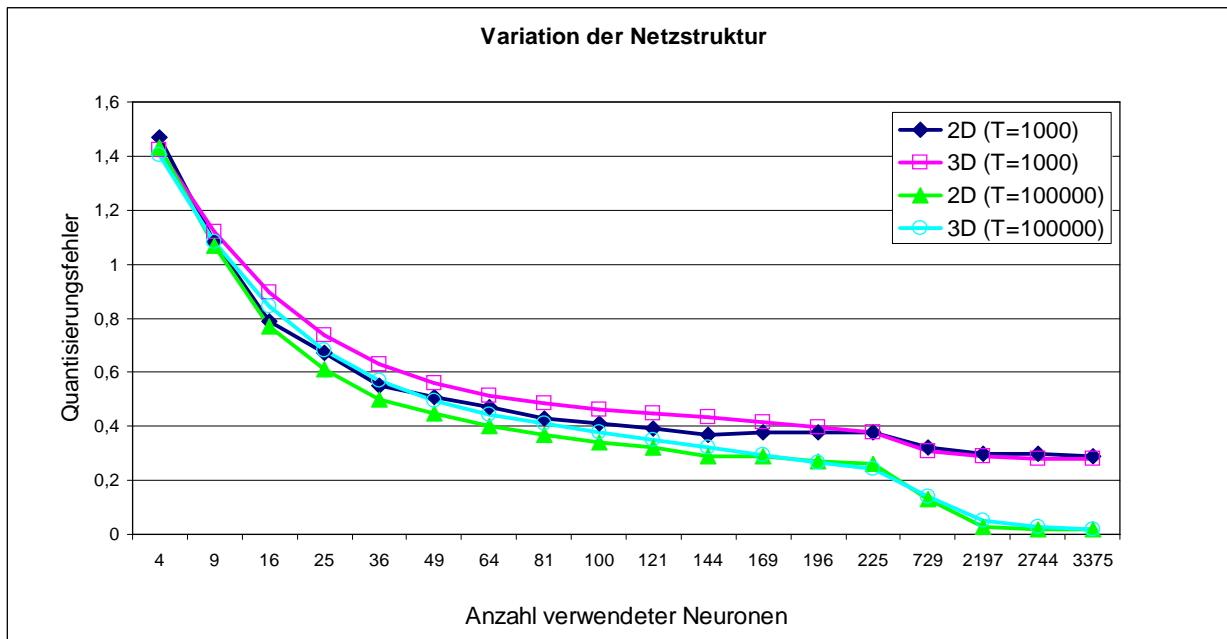


Abbildung 6.17: Analyse des Quantisierungsfehlers bei Variation der Netzstruktur und Neuronenanzahl

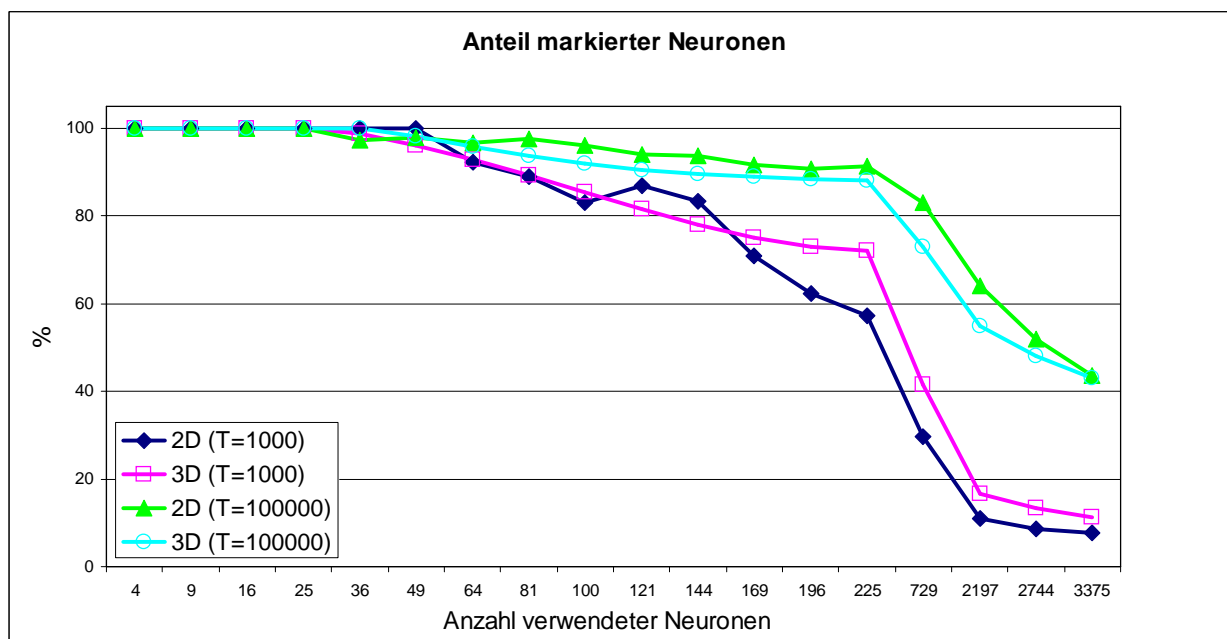


Abbildung 6.18: Verlauf des Anteils markierter Neuronen bei zunehmender Neuronenanzahl

## 6.2.5 Erkenntnisse

Zusammenfassend zeigt sich, dass der 3D-SOM-Fehleragent eine zuverlässigere Aussage über die Art bzw. den Zustand des akustischen Messsignals zulässt. Es werden deutlich weniger Anteile als unbekannt deklariert. Alle Mikrofon-signale, welche aus dem Innenraum eines fehlerfreien Fahrzeuges bei verschiedenen Fahrsituationen stammen, werden durchgängig mit über 50% als fehlerfrei vom 3D-SOM-Fehleragenten mit 6 Merkmalen erkannt. Wenn ein Defekt vorliegt, dann wurde eine

Messung mit der schlecht erkennbaren Fehlerklasse 9 (defekte Gelenkwelle, Außenlager rechts) zu über 40% (Abbildung 6.7) mit diesem Agenten als fehlerhaft eingestuft; bei Verwendung des 2D-SOM-Fehleragent mit gleicher Anzahl von Merkmalen jedoch nur zu 3% (Abbildung 6.6). Die besten Ergebnisse erreichte der 3D-SOM-Fehleragent ( $n=10$ ,  $T=200000$ ,  $M=6$ ). In Tabelle 6.8 sind die auftretenden Fehlergeräuschursachen und die Entscheidungsraten des Agenten (optimierte Parameter und Algorithmus Variante 1) dargestellt. Abgesehen von Klasse 9 und 10 werden alle von den Bauteilen emittierten auffälligen Geräuscharten über mindestens 83% der Gesamtzeit richtig als Fehlerklasse 1 (C1 - auffälliges Geräusch) erkannt. Das normale Fahrgeräusch (C2) - durch Fehlerklasse 12 abgebildet - wird über 87% der Gesamtzeit richtig durch den Agenten bewertet.

Fehlerart	Klasse	C1 in %	C2 in %	U in %
defektes Motorlager	1	95.0	0.3	2.4
Koppelstange ausgeschlagen rechts	2	94.6	1.3	1.6
Koppelstange ausgeschlagen links	3	90.9	1.1	2.1
Radlager stark beschädigt rechts	4	92.0	1.3	4.4
Radlager stark beschädigt links	5	87.0	2.9	3.6
defektes Radlager links	6	88.7	1.7	3.7
defektes Radlager rechts	7	93.4	0.6	2.6
Koppelstange lose geschraubt links	8	83.5	3.4	2.4
defekte Gelenkwelle Außenlager rechts	9	69.5	4.6	4.8
defektes Federbeinlager rechts	10	67.6	2.8	3.0
defektes Federbeinlager links	11	83.0	3.3	3.8
fehlerfrei/ Originalklang	12	1.8	87.3	3.4

Tabelle 6.8: Die mittleren Entscheidungsraten des 3D-SOM-Fehleragenten für 129 Mikrofonaufnahmen

Bei den Untersuchungen hat sich herausgestellt, dass der Anlernprozess bei einer Mittelung der Merkmale über 250 ms die besten Erkennungsraten erzeugt (siehe auch Kapitel 5.1). Bei der anschließenden Umsetzung des 3D-SOM-Agenten ist diese zeitliche Mittelung jedoch nicht notwendig, da die zeitkontinuierlichen Werte der Merkmale die auf den Mittelwert trainierten Neuronen aktivieren und zum anderen aus einer gepufferten Menge zufällig verteilt zur Bewertung herangezogen werden (siehe erster Absatz im Abschnitt 6.2.2).

## 6.3 Analyse der Hörbarkeit und Fehlerzustand-Erkennung

In Abschnitt 6.2.3 wurde die Anwendbarkeit des SOM-Agenten für die Erkennung und Unterscheidung eines akustisch auffälligen und unauffälligen Fahrzeuggeräusches anhand eines Innenraum-mikrofons nachgewiesen. Dabei hat sich gezeigt, dass der 3D-SOM Agent die besten Ergebnisse erreicht. Die Validierung der Anwendbarkeit auf die Unterscheidung der zwei Zustände - normales und auffälliges Fahrzeuginnenraumgeräusch - erfolgte durch die in Abschnitt 5.1 beschriebenen Datensätze, welche eine gute Hörbarkeit aufweisen. Die ausgewählten Mikrofonaufnahmen wurden bei entsprechender Parametrisierung des Agenten meist nur unter 10% der Analysezeit falsch bewertet (siehe 6.7). Im Folgenden wird das Testszenario weiter gestreut. Dies bedeutet, es werden

alle zur Verfügung stehenden Mikrofonaufnahmen durch den 3D-SOM-Agenten bewertet. Jede Mikrofonaufnahme wurde mit einer subjektiven Hörbarkeitsbewertung versehen. Diese erfolgte in 5 Schritten mit 0, 0.1, 0.3, 0.5, 0.8, 1, wobei eine gute bis sehr gute Hörbarkeit dem Faktor 0.8-1 entspricht. Treten vereinzelt hörbare Bereiche auf, so wurde die Aufnahme mit 0.5 bewertet. Bei schlechter bzw. nicht genau bestimmbarer Hörbarkeit des vom Fehlerbauteil verursachten Geräusches, wurde die Aufnahme mit 0.1-0.3 bewertet. Treten keine auffälligen Geräusche hervor, so gibt es die Bewertung 0.

### 6.3.1 Korrelation zwischen Hörempfinden und SOM-Agent-Ausgabe

Anhand dieser subjektiven Testreihe wird untersucht, wie gut die Unterscheidung des Geräusch-Zustandes durch den Agenten mit dem Empfinden des Menschen übereinstimmt. Da die Bewertung der Hörbarkeit nur durch eine kleine Anzahl von Personen durchgeführt wurde, kann diese nicht verallgemeinert werden und ist dementsprechend mit sehr weichen Grenzen zu beurteilen. Dennoch sollten sich Tendenzen ableiten lassen, ob zum Beispiel der Agent ein auffälliges Geräusch von einem defekten Bauteil nur dann erkennt, wenn dieses auch von einem Menschen deutlich wahrgenommen wird. Wie in Abbildung 6.19 gezeigt, wird die Grundlage durch eine Datenbank gebildet. Diese beinhaltet zustandsabhängige akustische Messdaten mit subjektiver Hörbarkeitsbewertung. Für den Trainingsprozess wurden, wie im vorigen Abschnitt 6.2.3 erläutert, nur die Mikrofon-signale zum Anlernen des Agenten benutzt, welche eine gute Hörbarkeit aufweisen. Das dadurch trainierte Wissen ist in den Gewichten des Neuronennetzes gespeichert. Im Validierungsprozess wird die Parametrisierung des Kohonennetzes geändert und die daraus resultierende Änderung des Entscheidungsverhaltens bewertet. Folgende Parametrisierung hat die besten Bewertungen erbracht:

- Typ: 3D
- Merkmale: 6
- Neuronen: 10
- Trainingsepochen: 20000

Die Ergebnisse sind in der Zusammenfassung unter 6.2.5 nachzulesen.

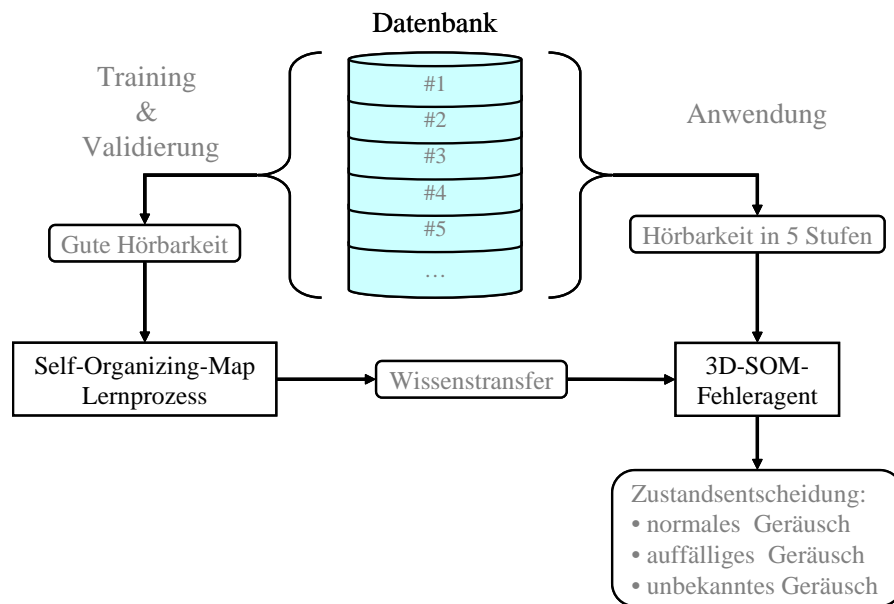


Abbildung 6.19: Schema zur Zustandsentscheidung durch den 3D-SOM-Fehleragenten

Die beste Adaption des Neuronennetzes wird als Grundlage für den im Schema dargestellten Bereich „Anwendung“ genutzt. Durch die Anwendung des Agenten auf alle in der Datenbank vorhandenen Datensätze - eingeteilt in fünf Hörbarkeitsstufen - wird eine neue Fehlerklassenanordnung erstellt. In Tabelle 6.9 sind die Fehlerarten und die dazu zugeordneten Klassen in der ersten und zweiten Spalte ersichtlich. Die Auswertung der Zustandsentscheidung des 3D-SOM-Agenten erfolgt für jede Hörbarkeitsstufe getrennt:

- 0 = nicht hörbar bzw. keine Auffälligkeit
- 0.1 = sehr vereinzelte Auffälligkeiten (unsicher Entscheidung)
- 0.3 = Auffälligkeiten (sichere Entscheidung)
- 0.5 = vereinzelt gut hörbare Auffälligkeiten
- 0.8 = gute Hörbarkeit
- 1 = sehr gute durchgängige Hörbarkeit

Für alle Mikrofonsignale jeder dieser Stufen kann dann durch die Mittelung der Zustandsentscheidungsraten eine Aussage über die Genauigkeit abgeleitet werden. Die Entscheidungsraten repräsentieren den prozentualen Anteil an der Gesamtzeit, für den der Ausgabewert des Agenten dem Fehlerzustand entspricht. Für die Hörbarkeit 0 sind die Raten in Tabelle 6.9 abgebildet. Es wurden 724 Innenraumaufnahmen durchgängig bewertet. Für die Klasse 2, 6 und 11 lagen keine Daten in der Datenbank (bei Hörbarkeit 0) vor.

Fehlerart	Klasse	C1 in %	C2 in %	U in %
Koppelstange links	0	77.55	8.07	7.73
Koppelstange rechts	1	79.82	5.20	8.40
Motorlager links	2			
Motorlager rechts	3	66.81	18.27	8.96
Radlager links	4	68.35	17.76	4.00
Radlager rechts	5	70.50	18.81	3.52
Antriebswelle links	6			
Antriebswelle rechts	7	37.56	46.00	5.88
Federbeinlager links	8	84.81	7.64	2.53
Federbeinlager rechts	9	78.08	8.15	4.92
Getriebelager links	10	71.98	13.68	6.19
Getriebelager rechts	11			
fehlerfrei	12	7.39	83.29	3.74

Tabelle 6.9: Entscheidungsraten des 3D-SOM-Fehleragenten für die Hörbarkeit 0 (724 Innenraumaufnahmen)

Bei der untersuchten Hörbarkeitsstufe werden alle Datensätze/ Aufnahmen in der Datenbank mit der Klasse 12 - fehlerfreier Fahrzeugzustand - untersucht, da diese selbstverständlich keine Auffälligkeiten aufweisen. In der Tabelle fällt auf, dass nur wenige Zeitabschnitte (2.53% - 8.96%) der Entscheidungsklasse „unbekannt“ zugeordnet wurden. Abgesehen von Klasse 7 werden alle Bauteil-Fehlerarten zu dem höchsten Prozentsatz in die Entscheidungsklasse „C1“ - fehlerbehafteter Zustand bzw. auffälliges Geräusch - eingeordnet. Die Einteilung in die Klasse „C2“ - unauffälliger Zustand - variiert zwischen 5.20% und 18.81%. Die Antriebswelle rechts (Klasse 7) stellt das einzige Bauteil dar, welches zu einem höheren Prozentsatz „C2“ zugeordnet wird. Alle dieser mit „C2“ bewerteten Messungen wurden auf Teerstraße bei Konstant- und Geradeausfahrt aufgenommen. Die Bewertung mit Hörbarkeit 0 erfolgte, weil absolut nichts außer das Fahrgeräusch zu hören ist. Der Fahrzustand „Kurvenfahrt“ wurde jedoch mit Hörbarkeit 1 bewertet, da dort die defekte Gelenkwelle durch den hohen Gelenkeinschlag eine hohe Belastung erfährt und Fehlergeräusche emittiert. Es kann also auditiv nachvollzogen werden, dass viele der analysierten Datensätze absolut keine hörbaren Auffälligkeiten aufweisen und die Entscheidung des Agenten dementsprechend richtig ist.

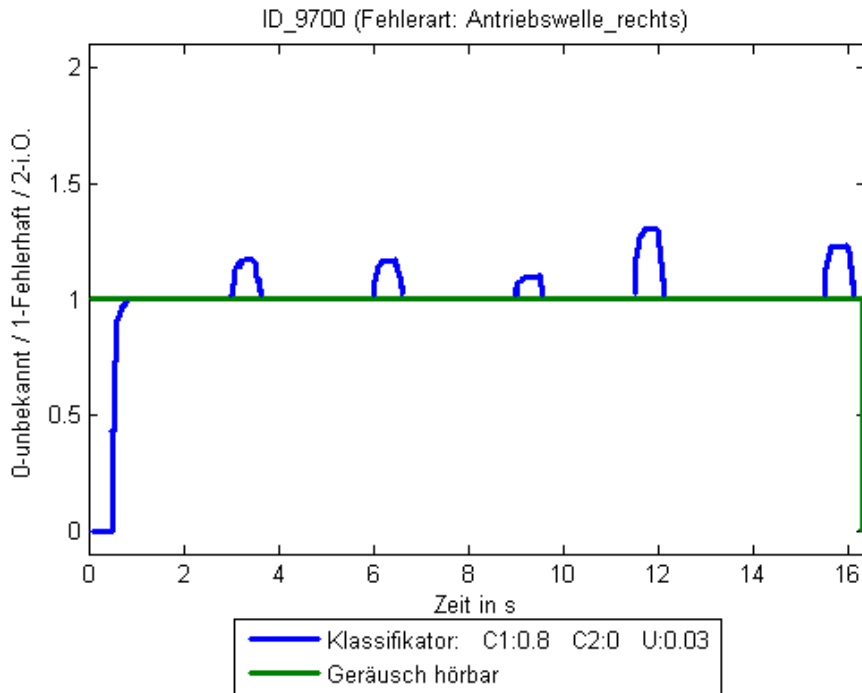
Fehlerart	Klasse	C1 in %	C2 in %	U in %
Koppelstange links	0	85.10	4.18	3.10
Koppelstange rechts	1	91.83	1.67	2.17
Motorlager links	2	90.25	0.75	1.75
Motorlager rechts	3	90.71	1.43	5.14
Radlager links	4	95.55	0.91	1.73
Radlager rechts	5	94.90	0.80	3.20
Antriebswelle links	6			
Antriebswelle rechts	7	54.16	12.95	12.32
Federbeinlager links	8			
Federbeinlager rechts	9	68.80	1.40	3.00
Getriebelager links	10			
Getriebelager rechts	11			
fehlerfrei	12			

Tabelle 6.10: Entscheidungsraten des 3D-SOM-Fehleragenten für die Hörbarkeit 1 (116 Innenraumaufnahmen)

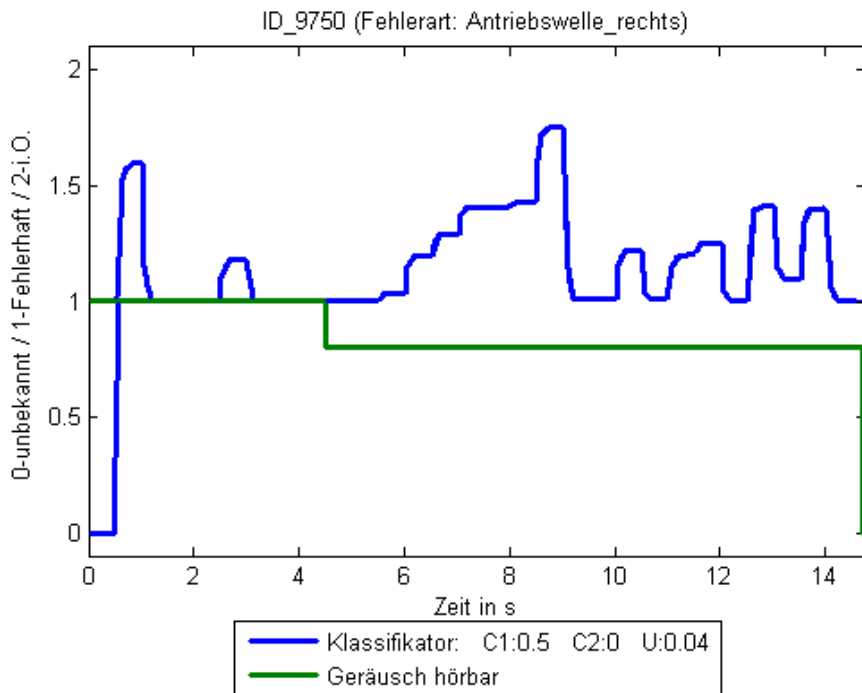
Dennoch kommt es in einigen Aufnahmen natürlich auch zu Fremdgeräuschen, wie z.B. durch Laptopklappen oder Bewegungsgeräuschen des Fahrers. Weiterhin kommt es auch zu Beschleunigungs-, Fahrbahn-, Lenk- und Geschwindigkeitsänderungen während der Aufnahme. Solche Änderungen sind in der Realität keine Seltenheit, stellen für den Agenten jedoch eine große Herausforderung dar. Dass der Agent-Entscheidungswert sich nicht statisch verhält, ist zu erwarten. Die meisten analysierten Messdaten haben eine Signallänge kleiner 40 Sekunden. Für die Generierung einer Fehlermeldung bzw. eines Fehlereintrags in einem Steuergerät wird ein deutlich größerer Beobachtungszeitraum nötig sein, um eine ausreichend sichere Aussage über den Zustand des Fahrzeugs abzuleiten. Die in Tabelle 6.10 gezeigten Entscheidungsraten bei einer Hörbarkeit von 1 zeigen deutlich weniger Streuung für die Zustände „C2“ und „U“, es werden also alle Aufnahmen - abgesehen von Klasse 7 und 9 - zu über 85% der Zeit als „auffälliges Geräusch“ vom 3D-SOM-Fehleragenten markiert. Die „fehlerhafte“ Zuordnung zu Klasse „C2“ erfolgt in weniger als 5% der Zeit, außer Klasse 7. Die Antriebswelle rechts wird als einziges Bauteil in nur knapp mehr als 50% der Zeit als fehlerbehaftetes Signal interpretiert. Es wurden 116 Innenraumaufnahmen durchgängig bewertet. In Abbildung 6.20 ist ein repräsentativer Verlauf für die Bewertung der Aufnahmen bei einer defekten rechten Antriebswelle dargestellt. Die Hörbarkeitsstufe für beide Messungen ist 1. Der Verlauf der Zustandsentscheidung für den Datensatz 9700 (rechte Antriebswelle defekt) wird in Bild 6.20a gezeigt. Die detaillierte Hörbarkeitsbewertung zeigt fast durchgängig eine Hörbarkeit von 1 (grüne Linie). Der Datensatz 9750 (ebenfalls rechte Antriebswelle defekt, Bild 6.20b) zeigt ab der 4. Sekunde eine geringere Hörbarkeit an, dementsprechend tendiert der Agent mehr in Richtung „fehlerfreier Zustand“. Es ist erkennbar, dass der Agent zwischen den Zuständen „C1“ und „C2“ wechselt. Dadurch sinkt die Entscheidungsrate von „C1“ über die Gesamtzeit. Ursachen hierfür können sein:

- schlechte Ausprägung des Geräusches (z.B. kleiner Lenkwinkel)
- seltene Anregung der Fehlergeräusch verursachenden Komponenten in den Bauteilen (z. B. Lager)

- unzureichende Relevanz der zum Training und Bewertung verwendeten akustischen Merkmale



a)



b)

Abbildung 6.20: Verlauf der Zustandsentscheidung für den Datensatz 9700 (Bild a) und 9750 (Bild b)

Im Folgenden soll der schwierige Fall der Hörbarkeitsstufe 0.1 analysiert werden. Diese Datensätze definieren sich durch vereinzelte Auffälligkeiten, jedoch kann bei diesen nicht eindeutig bestimmt werden, ob es sich dabei um ein defektes Bauteil handelt. Ein auffälliges Geräusch könn-



te dementsprechend auch durch ein Gurtschloss, Laptop, oder Fahrwerks- und Motorkomponenten verursacht werden.

Fehlerart	Klasse	C1 in %	C2 in %	U in %
Koppelstange links	0	66.17	25.22	4.33
Koppelstange rechts	1	80.53	11.44	3.76
Motorlager links	2			
Motorlager rechts	3	73.49	16.97	4.00
Radlager links	4	66.65	20.55	5.65
Radlager rechts	5	82.03	6.75	3.47
Antriebswelle links	6			
Antriebswelle rechts	7	15.30	64.30	11.10
Federbeinlager links	8	86.14	2.57	7.14
Federbeinlager rechts	9			
Getriebelager links	10	84.40	6.40	4.87
Getriebelager rechts	11			
fehlerfrei	12			

Tabelle 6.11: Entscheidungsraten des 3D-SOM-Fehleragenten für die Hörbarkeit 0.1 (227 Innenraumaufnahmen)

Wiederum sind die in Tabelle 6.11 gegebenen Entscheidungsraten für die Hörbarkeitsstufe 0.1 den Fehlerklassen zugeordnet. Es wurden 227 Innenraumaufnahmen durchgängig bewertet. Es zeigt sich erneut, dass die schwierig zu erkennende Antriebswelle nur zu einem sehr geringen Prozentsatz von 15.3% als fehlerhafter Zustand vom SOM-Agenten eingestuft wird. Zu 64.3% wird dieser jedoch als fehlerfreier Zustand eingeschätzt. Es ist auch ersichtlich, dass der Agent sich zu einem relativ hohen Anteil von 11.1% für den unbekannten Zustand entscheidet. Dies deutet darauf hin, dass vermehrt Werte aufgetreten sind, die entweder außerhalb des trainierten Eingaberaumes der Neuronenkarte des SOM-Agenten liegen oder keine eindeutig angeregten Neuronen aufgetreten sind. Das Training erfolgte durch ausgewählte Datensätze mit einer Hörbarkeitsstufe von 1, die Häufigkeitsverteilung über die 12 Zustände war gleich. Wenn das Ergebnis jedoch mit dem subjektiven Höreindruck verglichen wird, so bestätigt sich, dass die Antriebswelle nur äußerst selten und nicht eindeutig wahrnehmbar ist. Äquivalent zu Tabelle 6.9 ist auch ersichtlich, dass die anderen Fehlerzustände zu über 66-86% dem fehlerbehafteten Zustand zugeordnet werden. Die Zuordnung zum fehlerfreien Zustand variiert je nach Fehlerart zwischen 2% und 25%.

### 6.3.2 Fazit

In den Untersuchungen dieses Abschnitts ist gezeigt, dass bei guter Hörbarkeitsstufe (0.8-1) der 3D-SOM-Fehleragent von Fehlerbauteilen verursachte auffällige Innenraumgeräusche zuverlässig dem fehlerbehafteten Zustand (C1) zuordnet. Weiterhin wurde nachgewiesen, dass bei den Hörbarkeitsstufen 0 und 0.1 - wo eine eindeutige Wahrnehmung durch den Menschen kaum wahrnehmbar ist - der Agent für fast alle Fehlerarten auch noch zuverlässig funktioniert. Ausnahme ist die Antriebswelle, welche jedoch auch nur in sehr begrenzten Fahrsituationen gut hörbar ist. Als weitere Erkenntnis folgt, dass absolut fehlerfreie Geräusche auch als solche zu über 83% erkannt

wurden. Die tatsächliche Fehlentscheidungsquote des Agenten lag bei unter 8%. Bei Geräuschen, die kontinuierlich auftreten, kann der Agent auch bei starker Überdeckung (z.B. Anregung Kopfsteinpflaster) eine zuverlässigere Aussage liefern, ob ein fehlerhaftes Geräusch vorliegt, als die zu Grunde liegende Hörbarkeitsbewertung des Menschen. Für eine endgültige sichere Entscheidung, ob ein Geräusch fehlerhaft oder fehlerfrei bewertet wird, sollte das Verhältnis zwischen den Entscheidungsraten analysiert werden:

- sichere Entscheidung: Differenz zwischen den 3D-SOM Entscheidungsraten für C1 und C2 ist größer 50%
- unsichere Entscheidung: Differenz zwischen den 3D-SOM Entscheidungsraten für C1 und C2 ist kleiner 50%

Dabei sollten die Entscheidungsraten für C1 und C2 über einen Zeitraum größer 5-10 Sekunden ermittelt werden, damit eine ausreichende, statistische Sicherheit vorliegt.

## 7 Akustisches Diagnose-System

Neben den Aufgaben der vorigen Kapitel im Bereich Datenbankaufbau, *Data Mining*, Algorithmenentwurf und Simulation steht bei der Bearbeitung auch die Entwicklung eines Messsystems für den Einsatz in Forschung, Entwicklung und Qualitätssicherung im Vordergrund. Dafür werden in diesem Kapitel zuerst die Anforderungen erläutert und anschließend die daraus entstandenen Systeme vorgestellt.

### 7.1 Motivation, Anforderungen und Umsetzung

Prinzipiell lassen sich zwei maßgebende Entwicklungsschritte aufzählen:

- Für den Einsatz in der Werkstatt wurde das *integrierte Mikrofonarray-Diagnose-System* (iMAD-System) entwickelt. Dieses dient als Werkzeug zur Erfassung von Geräuschen bei Kundenbeanstandungen mit anschließendem Austausch und Datensicherung im Bereich der Qualitätssicherung.
- Für den Einsatz in Forschung und Entwicklung wurde das *mobile akustische Diagnose-System* (mADS) entwickelt. Dieses dient zur Geräuschanalyse und Anwendung von neuen Algorithmen in Versuchsträgern oder am Bandende.

Wie auch in [Put09] erläutert ist, besteht ein Interesse der Qualitätssicherung darin, den Austausch sowie die Fehlerbehebung bei akustischen Beanstandungen zu verbessern und zu vereinfachen. Dazu sollen Werkstatt und Qualitätssicherung eine einheitliche Sprache sprechen, wenn auffällige Geräusche am Kundenfahrzeug zu analysieren sind. Für die dafür notwendige Geräusch- und Protokollerfassung wurde das iMAD-System entwickelt.

Im Rahmen der Forschungstätigkeiten bestand der Wunsch, möglichst flexibel und handlich Geräusche erfassen zu können und mit den neuesten Algorithmen und Modellen direkt im Fahrzeug zu experimentieren bzw. Simulationsergebnisse zu validieren. Dafür wurde das mADS entwickelt.

### 7.2 iMAD - Geräuscherfassung mit dem Diagnosetester

In ersten Untersuchungen zum Thema Geräuscherfassung im Fahrzeuginnenraum wurden verschiedene Mikrofonkapseln als ein- und mehrkanalige Aufnahmeeinheit untersucht. Mehrkanalige Mikrofoneinheiten bieten den Vorteil einer virtuellen Richtcharakteristik (siehe [Ker08]), benötigen dafür jedoch mehr Platz im Fahrzeuginnenraum. Basierend aus diesen Analysen haben sich zum einen der Name des Systems, sowie folgende zwei Punkte ergeben:

- Bereits mit Mikrofonkapseln im unteren Preissegment (1-2€ Ladenpreis) lassen sich sehr gute Aufnahmen erzielen. Die Mehrkosten für eine *Highend*-Mikrofonkapsel stehen für den Anwendungsfall der Inneraumgeräuscherfassung nicht in Relation zu der etwas schlechteren Richtcharakteristik und dem minimal niedrigeren Rauschabstand der preiswerteren Kapsel. Einige Messungen und Eigenschaften der untersuchten Mikrofonkapseln sind in Anhang A.2 dargestellt.
- Eine mehrkanalige Anwendung und die Verwendung von Algorithmen zur Schallquellenlokalisierung (beispielsweise *Beamforming*, *MUSIC*) bringen nur unbefriedigende Ortungsergebnisse in einem Frequenzbereich unter 1000Hz. Dies liegt zum einen an dem stark diffusen Schallfeld, dem teilweise sehr schwach ausgeprägten Geräuschquellen sowie an korrelierten Quellen.

Für ein Geräuscherfassungssystem basierend auf den VAS-Diagnosetestern (diese werden in den Werkstätten zur Analyse der Fahrzeugfunktionen eingesetzt) wurde deshalb nur ein Mikrofon verwendet, welches aus einer Mikrofonkapsel im unteren Preissegment und einer Signalvorkonditionierung besteht. Die VAS-Diagnosetester (z.B. VAS 5052a, siehe Abbildung 7.1) werden in den Werkstätten bereits verwendet und sind deshalb als Entwicklungsplattform sehr gut geeignet.



Abbildung 7.1: VAS 5052a Diagnosetester mit iMAD-Software (links) und mehrkanalige Mikrofoneinheit (rechts)

Die entwickelte Software enthält Funktionen zur Aufnahme, Protokollerfassung, Analyse und Sicherung, wobei die Analysefunktionen (z. B. Frequenzbereichsbetrachtung) für den Werkstattgebrauch nicht freigeschaltet sind. Lediglich ein Abhören der bereits aufgenommenen oder gerade laufenden Messung wird gestattet. Dadurch soll die Bedienfreundlichkeit erhöht, die Komplexität verringert und der Werkstattbetrieb schnell und reibungslos gestaltet werden. Die Entwicklung der Software wird im Wesentlichen durch die Hardware definiert. Beispielsweise durch ältere Diagnosetester, welche nur USB 1.1 als Kommunikationsschnittstelle aufweisen. Das Mikrofon besitzt eine eingebaute Soundkarte und ermöglicht dadurch eine hohe Betriebssicherheit der Software auf verschiedenen Systemen. Die aufgenommenen Geräusche werden über bereits bestehende Kommunikationssys-

teme an die Qualitätssicherung weitergeleitet und dort abgelegt und analysiert (siehe [Put09]).

### 7.3 mADS - mobiles akustisches Diagnosesystem

Das mADS stellt ein Aufnahme- und Analysesystem für die Forschung und Entwicklung dar. Auf Basis dieses Systems können Geräusche, Schwingungen, Fahrzeugdaten und analoge bzw. digitale Signale erfasst und verarbeitet werden. Diese Daten dienen als Grundlage für die Umsetzung und den Test verschiedenster Algorithmen - vorwiegend der akustischen Diagnose - für unterschiedliche Konzernbereiche. Dabei können diese Algorithmen direkt in die mADS Software integriert (in C# oder Matlab®) werden und dadurch der Funktionstest erfolgen.

Die Vorgaben für das System wurden wie folgt definiert:

- flexibel und handlich, kompakte Abmessungen
- handelsüblicher Laptop mit Windows XP
- Aufnahmesystem mit frei wählbarer Anzahl von Kanälen für beispielsweise Mikrofone, Beschleunigungsaufnehmer und Kunstkopf
- CAN-Bus-Schnittstelle
- Analysemöglichkeiten: alle Parameter aus Kapitel 4.1 und 4.2



Abbildung 7.2: mADS-Hardware

Wie in Abbildung 7.2 gezeigt, wurde das System nach den Vorgaben aus folgenden Komponenten zusammengestellt. Es besteht aus Laptop mit mADS-Software und Hardware, welche von der Fa. *HeimSystems* unter dem Namen *DataRec 4* verkauft wird. Es handelt sich dabei um ein modulares

System zur Erfassung und Konditionierung von diversen Signalen. In [Gmb07] werden sowohl die Haupteigenschaften zusammengefasst als auch die einzelnen Module vorgestellt.

Wesentliche *DATArec 4* Eigenschaften:

- Modulares, kompaktes und robustes Design für mobile Anwendungen mit quasi unbegrenzten Konfigurationsmöglichkeiten
- IEEE1394b, USB 2.0, Ethernet
- Module (galvanisch getrennt) mit integrierter Signalkonditionierung (Mikrofone, Dehnungsmessstreifen, Beschleunigungs- und Drucksensoren, ICP usw.)
- Spezielle Module für digitale Daten (AES/EBU, CAN, usw.)
- Digitale Signalverarbeitung für höchste Genauigkeit (24 Bit A/D Wandler, 100dB Dynamik;  $<0.2^\circ$  Phasenfehler)

Für das mADS-System wurde das Power-Modul zur Spannungsversorgung, das Signal-Modul zur Kommunikation über Ethernet und das DEBU-Modul zur Signalerfassung verwendet. Wobei letzteres zwei digitale Eingangskanäle (z.B. für den Kunstkopf) und 4 analoge/ICP Eingangskanäle zur Verfügung stellt. Die Abtastrate der analogen Kanäle kann im Bereich 800 Hz bis 200 kHz gewählt werden. Der 24 Bit Analog-digital-Wandler realisiert einen Dynamikbereich von  $>102$  dB. Der Signalpegel liegt bei  $\pm 10 mV_{peak}$  bis  $\pm 10 V_{peak}$ . Weitere technische Daten aller Module sind in [Gmb07] zusammengefasst.

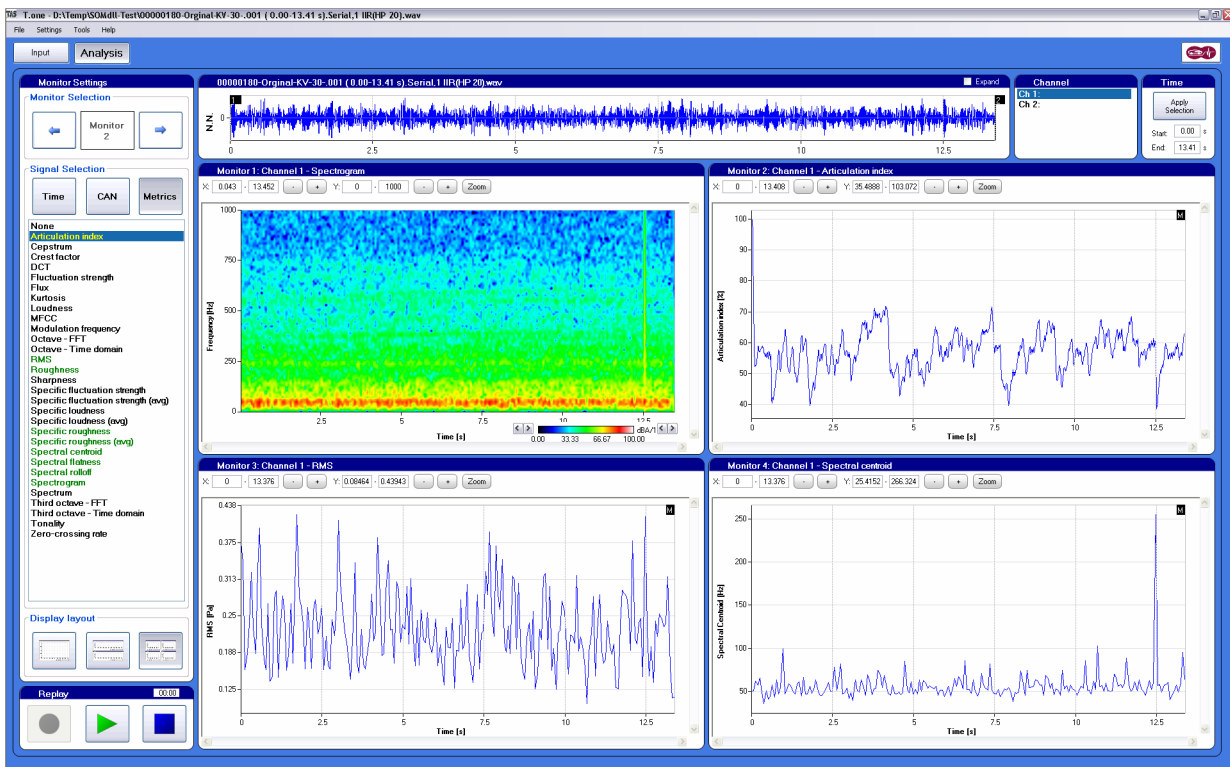


Abbildung 7.3: Die Software des mADS bietet diverse Funktionen zur Signalanalyse, Protokollerrfassung, Wiedergabe und Berechnung eigener Algorithmen.

In Abbildung 7.3 ist die Analyse-Oberfläche der mADS-Software, welche in C# (*C-Sharp*, eine von Fa. Microsoft entwickelte objektorientierte Programmiersprache) entwickelt wurde, dargestellt.

Im linken Bereich kann die zu visualisierende Analysefunktion ausgewählt werden. Dabei können wahlweise die Zeitsignale der Sensoren, ausgewählte Fahrzeug-CAN-Bus-Nachrichten oder Kenngrößen (z.B. psychoakustische wie Schärfe oder Lautheit) auf bis zu vier Monitoren abgebildet und analysiert werden. Entwickelte Algorithmen können über eine in Matlab® generierte C# kompatible DLL (*Dynamic Link Library*<sup>1</sup>) in der mADS-Software verwendet werden. In Abbildung 7.4 ist das Menü zum Starten der Matlab®-DLL gezeigt. Es werden im mittleren Fenster die zu verwendenden Merkmale, Fenstergröße und Mittelung definiert und anschließend die Berechnung des SOM-Agenten wahlweise im *sample*- oder *block*-Modus berechnet. Letzteres definiert, ob jeder Berechnungswert einzeln oder alle Berechnungswerte in einem Block an die DLL übertragen werden. Das linke und rechte Fenster zeigen die Ausgabe der Matlab®-DLL, zum einen den aktuellen Berechnungsstatus (als *Figure 1* gekennzeichnet) und zum anderen den Verlauf des Ausgabewerts des SOM-Agenten (als *Figure 2* gekennzeichnet, siehe auch Abschnitt 6.2).

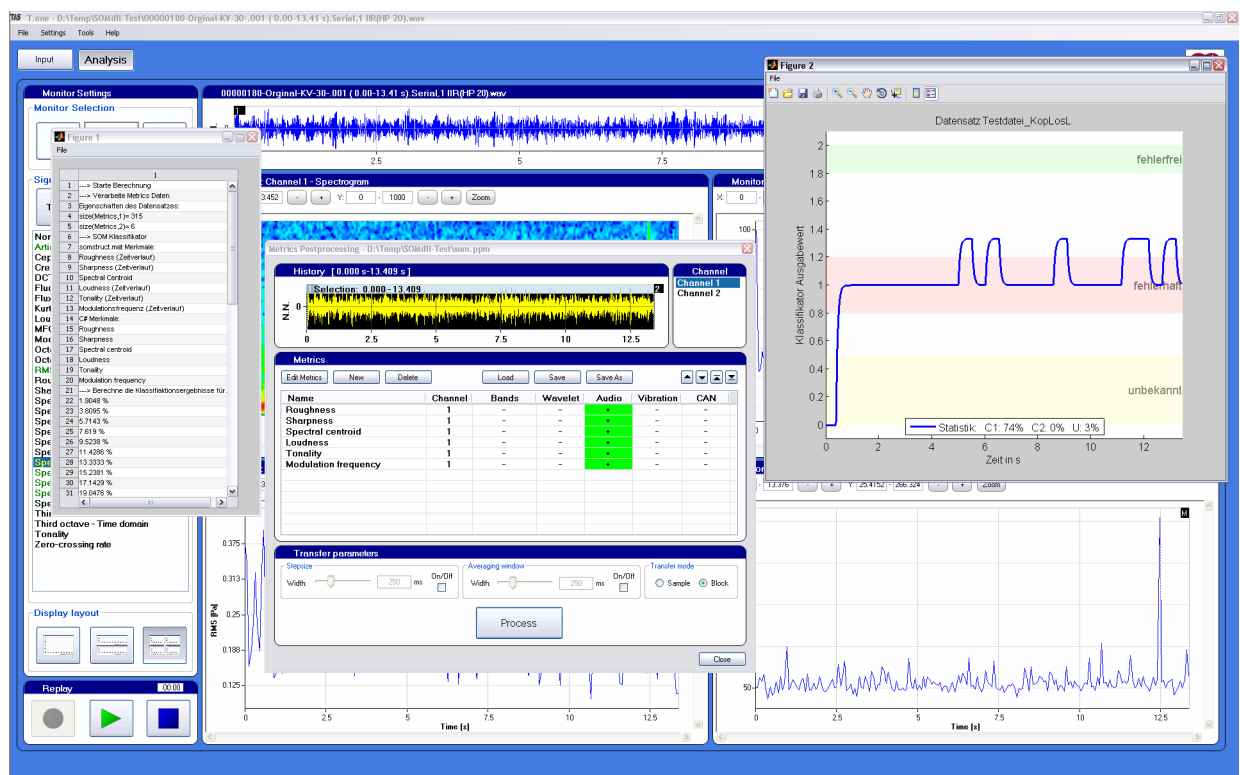


Abbildung 7.4: Aufruf der Matlab®-DLL aus der mADS-Software

Weiterhin wurden Schnittstellen zu einer SQL-Datenbank entwickelt und die Struktur der in Abschnitt 3.1 Abbildung 3.2 gezeigten Datenablage implementiert. Dadurch können Aufnahmen in dieser Datenbank direkt abgelegt werden. Der Zugriff auf die SQL-Datenbank kann auch über Matlab® erfolgen, wodurch Algorithmenentwicklung und Simulation auf realen Messdaten erfolgen können.

<sup>1</sup>Eine Benutzung der DLL ist auch ohne Matlab®-Lizenz durch das *Matlab-Component-Runtime* Paket möglich





## 8 Zusammenfassung & Ausblick

Die Vision einer intelligenten Akustikdiagnose ist die Überwachung und selbstständige Analyse des Fahrzeugklangs im Innenraum auch während der Nutzungsphase durch den Kunden. Um dieses Ziel zu erreichen, stellt die wichtigste Aufgabe der Aufbau einer Datenbank und der damit verbundene Zugriff auf eine Vielfalt an Fahrzeuggeräuschen dar. Als Forschungsschwerpunkt sind zwei Fahrzeuge mit einem handelsüblichen Messsystem in verschiedensten Fahrsituationen vermessen worden, um eine Vielzahl an Versuchsdaten zu erfassen. Das Volkswagen-Geräuschlexikon stellt eine erste Grundlage zur Erfassung von Geräuschen an Kundenfahrzeugen dar. Auf Basis eines Diagnosetesters, dieser wird in den Werkstätten eingesetzt, ist das *integrierte Mikrofonarray-Diagnose-System* (iMAD-System) zur Geräuschaufnahme realisiert. In Zusammenarbeit mit der Qualitätssicherung ist ein Einsatz dieses Systems in den Werkstätten zur Erfassung von akustischen Kundenbeanstandungen geplant, um in Zukunft die Geräuschdatenbank als Diagnosehilfe einsetzen zu können.

Der Aufbau einer ersten Forschungsdatenbank untermauerte, dass kein Softwarepaket die Anforderungen zur Aufnahme, Protokollierung und Berechnung klassischer akustischer und psychoakustischer Kenngrößen erfüllt. Aus den definierten Anforderungen wurde deshalb das *mobile akustische Diagnose-System* (mADS) entwickelt. Die Ableitung eines akustischen Fingerabdrucks aus den berechneten akustikbeschreibenden Kenngrößen ermöglicht eine Erkennung von Fehlergeräuschen direkt aus den Mikrofonsignalen.

Zur Identifikation eines akustischen Fingerabdrucks ist eine Datenbank, welche die Zeitsignale der Mikrofone im Fahrzeuginnenraum, die Messbeschreibung sowie berechnete Kenngrößen enthält, genutzt worden. Insgesamt sind für ein definiertes Zeitfenster der Länge  $t(n)$  eine Vielzahl an Kenngrößen (diese werden als Merkmale im Klassifikationsprozess verwendet) berechnet worden. Eine kleinere Zeitkonstante  $t(n)$  bewirkt für die Berechnung der Kenngrößen eine genauere akustische Abbildung der Fehlerklassen und somit ein besseres Klassifikationsergebnis. Der Zielkonflikt zwischen dem Berechnungsaufwand für eine große Menge an Merkmalen und einer möglichst hohen Erkennungsrate des Mustererkenners wurde durch Verfahren zur Reduktion der Dimensionen gelöst. Dabei konnte eine Reduktion auf eine überschaubare Anzahl an Merkmalen bei gleichbleibend guter Erkennungsrate erreicht werden. Als besonders effektiv hat sich eine kaskadierte Verarbeitung erwiesen. Diese besteht aus einer Merkmalsbewertung durch die Fisher-Diskriminante mit nachgeschalteter Algorithmik zur Rangordnungserzeugung. Automatische Verfahren zur Rangberechnung wie Transinformation, *Multi-Relief* und *Fast Correlation Based Filter* erreichten die höchsten Klassifikationsraten. So konnten bereits bei 9 durch den *Multi-Relief* (bei einer vergleichbar geringen Rechenzeit) ermittelten Merkmalen Erkennungsraten größer 90 % für die Trennung von 12 verschiedenen Geräuscharten erreicht werden. Eine minimale Reduzierung der Erkennungsrate um ca. 5 bis 10 Prozent erfolgte bei nur 6 verwendeten Merkmalen.

In einem weiteren Schritt wurde analysiert, inwieweit eine Fahrzeugabhängigkeit in den Merkmalen vorliegt. Es steht fest, dass der Fahrzeugtyp einen entscheidenden Einfluss auf die Merkmalausprägung ausübt. Jedoch wurde anhand der Untersuchungen an verschiedenen Fahrzeugen bestätigt, dass ein universeller akustischer Fingerabdruck ähnlich hohe Erkennungsraten erreicht wie ein fahrzeugspezifischer Fingerabdruck. Eine Umsetzung der Berechnungsalgorithmen ist nach Meinung des Autors für zukünftige Anwendungen im Fahrzeug für alle Fahrzeugtypen gleich, nur eine Adaption der Wertebereiche ist notwendig. Die Suche nach verallgemeinerten Merkmalen, die nicht durch den Fahrzeugtyp beeinflusst werden, stellt eine Aufgabe weiterer Forschung dar. Dabei sollte auch untersucht werden, ob anhand der in dieser Arbeit generierten Fingerabdrücke eine Allgemeingültigkeit durch ein modifiziertes Klassifikationsmodell erreicht werden kann. Bisher wurden die Merkmale zeitinvariant durch den Klassifikator bewertet, da deren Berechnung gemittelt über das Zeitfenster  $t(n)$  durchgeführt wurde. Der charakteristische Zeitverlauf eines Geräusches wird dabei vernachlässigt, dieser stellt jedoch ebenfalls eine Informationsquelle dar. So ist es einem geschulten Menschen möglich, ein defektes Radlager unabhängig vom Fahrzeugtyp nicht zuletzt anhand der zeitlichen Änderung des Geräusches zu erkennen. Diese zeitliche Änderung der Merkmale wurde bisher nicht im Klassifikationsmodell berücksichtigt.

Eine weitere bearbeitete Aufgabe ist die automatische Erkennung von auffälligen Geräuschen anhand des akustischen Fingerabdrucks. In den bisher genannten Untersuchungen sind Geräusche anhand einer durch den Menschen gewählten Sequenz klassifiziert worden. Der Fokus richtet sich nun darauf, ob ein Algorithmus in der Lage ist, auffällige Geräusche, die von verschlissenen oder defekten Bauteilen emittiert werden, automatisch anhand einer Fahrzeuginnenraumaufnahme zu erkennen. Anhand eines Modells, der sogenannte SOM-Fehleragent, ist erkannt worden, dass eine Erkennung von defekten Geräuschen möglich ist. Die Validierung erfolgte auf einer Vielzahl von Fahrzeuggeräuschen. Für nahezu alle untersuchten Fehlergeräusche konnten, je nach Auslegung des Modells, sehr gute Erkennungsraten erreicht werden. Parameteruntersuchungen des Modells zeigten eindeutige Abhängigkeiten von der Anzahl der verwendeten Neuronen. Die Merkmale des akustischen Fingerabdrucks wurden gemittelt berechnet und dem Agenten zeitunabhängig zugeführt, so dass sich die Charakteristik des Fahrzeugs sehr stark in einer unterschiedlichen Ausprägung der Merkmale bei gleichem Geräusch widerspiegelt. Einen bedeutenden Einfluss zeigt der Aufwand zur Berechnung der jeweiligen Merkmale. Mehrwertige Merkmale, wie beispielsweise das Spektrum über das Zeitfenster  $t(n)$ , benötigen einen deutlich höheren Rechenaufwand als der einwertige Effektivwert. Der SOM-Fehleragent ist aus diesen Gründen mit möglichst wenigen (beispielsweise 6) und einfach zu berechnenden einwertigen Merkmalen getestet worden. Der Nachteil einer fahrzeugabhängigen Erweiterung der Lerndatenmenge wird deshalb durch Recheneffizienz kompensiert. Der SOM-Fehleragent kann so mit wenigen einwertigen Merkmalen Auffälligkeiten erkennen und im nächsten Schritt hinsichtlich des emittierenden Bauteils klassifizieren. Zukünftige Untersuchungen sollten die zeitliche Veränderung der Geräusche mit berücksichtigen und dadurch eine weitere Reduktion der zu verwendenden Merkmale und damit eine fahrzeugunabhängige Klassifikation erreichen.

Hervorzuheben sind auch die Untersuchungen zum Einfluss der Hörbarkeit. Eine prinzipielle Frage ist, ob ein Agent zur Geräuscherkennung, ähnlich des Menschen, in Abhängigkeit der Geräuschausprägung eine unterschiedliche Erkennungsrate erreicht. Für diese Analysen sind die

---

Fahrzeuggeräusche in verschiedene Hörbarkeitsstufen eingeteilt worden. Anschließend erfolgte ein Vergleich zwischen den Erkennungsraten und der Hörbarkeit der klassifizierten Geräusche. Bei guter Hörbarkeit kann das von den Fehlerbauteilen verursachte auffällige Innenraumgeräusch zuverlässig durch den SOM-Fehleragenten dem fehlerbehafteten Zustand zugeordnet werden. Bei schlechter Hörbarkeit, wo eine eindeutige Wahrnehmung durch den Menschen kaum möglich ist, funktioniert der Fehleragent für fast alle Fehlerarten zuverlässig, jedoch mit geringerer Erkennungsrate.

Die akustische Diagnose ist ein Gebiet von Interesse in der Entwicklung und Qualitätssicherung, aber viele Grundlagen für eine automatische, fortwährende Analyse des Kundenfahrzeugs sind bisher noch nicht gegeben. Es sind jedoch aufgrund der in dieser Arbeit durchgeführten Untersuchungen, Entwicklungen, Absprachen und Planungen bereits neue Projekte mit Abteilungen der Qualitätssicherung und Entwicklung bei Volkswagen initiiert, die eine Bearbeitung der Geräuscherfassung am Kundenfahrzeug, deren Ablage, Verarbeitung und zukünftige Diagnosefunktionen, beispielsweise für das Soundsystem ermöglichen sollen.



# Literaturverzeichnis

- [AD91] ALMUALLIM, HUSSEIN und THOMAS G. DIETTERICH: *Learning With Many Irrelevant Features*. In: *In Proceedings of the Ninth National Conference on Artificial Intelligence*, Seiten 547–552. AAAI Press, 1991.
- [AFHN08] ABRAMOVICI, MICHAEL, MADJID FATHI, ALEXANDER HOLLAND und MANUEL NEUBACH: *Integration von Feedbackdaten aus der Produktnutzungsphase im Rahmen des PLM-Konzepts*. In: *Multikonferenz Wirtschaftsinformatik*, 2008.
- [AN07] ASUNCION, A. und D.J. NEWMAN: *UCI Machine Learning Repository*, 2007.
- [Aur85a] AURES, W.: *Berechnungsverfahren für den sensorischen Wohlklang beliebiger Schall-signale*. In: *Acustica 59*, Seiten 130–141, 1985.
- [Aur85b] AURES, W.: *Ein Berechnungsverfahren der Rauigkeit*. In: *Acustica 58*, Seiten 268–281, 1985.
- [Aya99] AYAN, N. F.: *Using information gain as feature weight*. In: *Proceedings of the 8th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN'99)*, 1999.
- [Bau97] BAUER, SEBASTIAN: *Fehlerdiagnose bei Luftfahrttriebwerken*. Fortschritt-Berichte VDI, 1997.
- [BBR08] BRUGGER, D., M. BOGDAN und W. ROSENSTIEL: *Automatic Cluster Detection in Kohonen's SOM*. *Neural Networks, IEEE Transactions on*, 19(3):442–459, 2008.
- [BD05] BIESIADA, J. und W. DUCH: *Feature Selection for High-Dimensional Data: A Kolmogorov-Smirnov Correlation-Based Filter*. *Advances in Soft Computing*, Springer Verlag, 0:95–104, 2005.
- [Ber47] BERANEK, L. L.: *The Design of Speech Communication Systems*. In: *Proceedings of the IRE*, Band 35, Seiten 880–890, 1947.
- [Ber08] BERTHOLD, TIM: *Algorithmenentwicklung zur Transferpfadortung mittels Blinder Quell-entrennung*. Diplomarbeit, Leibniz Universität Hannover, 2008.
- [BFOS84] BREIMAN, LEO, JEROME H. FRIEDMAN, RICHARD A. OLSHEN und CHARLES J. STONE: *Classification and Regression Trees*. Chapman & Hall, New York, NY, 1984.
- [BJGS01] BAILER-JONES, C. A.L., R. GUPTA und H. P. SINGH: *An introduction to artificial neural networks*. Technischer Bericht, Max-Planck-Institut für Astronomie, IUCAA, Department of Physics Sri Venkateswara College, 2001.
- [BL02] BARBALATA, C. und L. LEUSTEAN: *Average monthly liquid flow forecasting using neural networks*. Technischer Bericht, Marin Preda High School, National Institute for Research and Development in Informatics-ICI, 2002.

- [Brü10a] BRÜEL & KJAER: *Application Note: Psychoacoustic Test Bench BZ 5301*, Brüel & Kjaer, 2010.
- [Brä10b] BRÄUER, J.: *Entwicklung eines evolutionären Algorithmus zur aktiven Akustikgestaltung von Elektrofahrzeugen*. Diplomarbeit, TU Berlin, 2010.
- [BTR10] BRÄUER, J., S. TWIEG und M. RÖTTING: *Evolutionary algorithms for the engine sound specification of electrical vehicles*. In: *Fortschritte der Akustik, DAGA 2010*, 36. Deutsche Jahrestagung für Akustik, 2010.
- [Bur98] BURGESS, CHRISTOPHER J.C.: *A Tutorial on Support Vector Machines for Pattern Recognition*. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.
- [BWJ06] BO, LIEFENG, LING WANG und LICHENG JIAO: *Multi-layer Perceptrons with Embedded Feature Selection with Application in Cancer Classification*. *Chinese Journal of Electronics*, 15:832–835, 2006.
- [Caw00] CAWLEY, G. C.: *MATLAB Support Vector Machine Toolbox (v0.50)*. University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000.
- [Cha05] CHAPPELLE, OLIVIER: *Feature selection for Support Vector Machines*. Technischer Bericht, Max Planck Institute for Biological Cybernetics, Tübingen, 2005.
- [Cho01] CHOUCHOULAS, ALEXIOS: *Incremental Feature Selection Based on Rough Set Theory*. Doktorarbeit, The University of Edinburgh, 2001.
- [Cin03] CINAR, M. SERHAT: *Neuronale Netze und Musik*. Technischer Bericht, Fachhochschule Köln, 2003.
- [CLHW06] CHENG, GUOJIAN, TIANSHI LIU, JIAXIN HAN und ZHENG WANG: *Towards Growing Self-Organizing Neural Networks with Fixed Dimensionality*. In: *WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*, Band 22, Seiten 157–162, OCTOBER 2006. ISSN: 2070-3724.
- [CT93] CHAPPELL, GEOFFREY J. und JOHN G. TAYLOR: *The temporal Kohonen map*. *Neural Netw.*, 6(3):441–445, 1993.
- [DG] DEMO-GNG: *Growing Self-Organizing Networks Java-Demo*, Institut für Neuroinformatik an der Ruhr-Universität Bochum. <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html>.
- [DH99] DER, RALF und MICHAEL HERRMANN: *Second-Order Learning in Self-Organizing Maps*. In: *In Kohonen Maps*, Elsevier Science, Seiten 293–302. Elsevier, 1999.
- [DP03] DING, CHRIS und HANCHUAN PENG: *Minimum Redundancy Feature Selection from Microarray Gene Expression Data*. In: *CSB '03: Proceedings of the IEEE Computer Society Conference on Bioinformatics*, Seite 523, Washington, DC, USA, 2003. IEEE Computer Society.
- [DP05] DING, CHRIS H. Q. und HANCHUAN PENG: *Minimum Redundancy Feature Selection from Microarray Gene Expression Data*. *J. Bioinformatics and Computational Biology*, 3(2):185–206, 2005.

- [Edl01] EDLER, ANDREAS: *Nutzung von Felddaten in der qualitätsgetriebenen Produktentwicklung und im Service*. Doktorarbeit, Fakultät V (Verkehrs- und Maschinensysteme), Technische Universität Berlin, 2001.
- [Fin03] FINK, GERNOT A.: *Mustererkennung mit Markov-Modellen*. Leitfäden der Informatik. Teubner B.G. GmbH, Stuttgart – Leipzig – Wiesbaden, October 2003.
- [Fis36] FISHER, R.A.: *The Use of Multiple Measurements in Taxonomic Problems*. Annals of Eugenics, 7:179–188, 1936.
- [FLB08] FINCKE, TONIO, VICTOR LOBO und FERNANDO BACO: *Visualizing self-organizing maps with GIS*. In: *GI Days 2008*, Münster, Germany, 2008.
- [Fle05] FLENTGE, FELIX: *Aktionenlernen mit Selbstorganisierenden Karten und Reinforcement Learning*. Doktorarbeit, Johannes Gutenberg-Universität Mainz, 2005.
- [FWH98] FOCHEM, M., P. WISCHNEWSKI und R. HOFMEIER: *Installation akustischer Prüfsysteme in die Fertigungsendkontrolle*. Technischer Bericht, MIT - Management Intelligenter Technologien GmbH, 1998.
- [FYH<sup>+</sup>08] FEENSTRA, K. ANTON, KAI YE, JAAP HERINGA, ADRIAAN P. IJZERMAN und ELENA MARCHIORI: *MultiRELIEF: specificity-determining residues from alignments by Machine Learning and Feature Weighting*. In: *Bioinformatics*, 2008.
- [FZ07] FASTL, HUGO und EBERHARD ZWICKER: *Psychoacoustics: facts and models*. Springer Verlag, 2007.
- [GBNT04] GILAD-BACHRACH, RAN, AMIR NAVOT und NAFTALI TISHBY: *Margin based feature selection - theory and algorithms*. In: *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, Seite 43, New York, NY, USA, 2004. ACM.
- [GE03] GUYON, ISABELLE und ANDRÉ ELISSEEFF: *An introduction to variable and feature selection*. J. Mach. Learn. Res., 3:1157–1182, 2003.
- [Gei08] GEISS, MARKUS: *Systemidentifikation von Fahrzeugstrukturen hinsichtlich ihres Körperschallübertragungsverhaltens*. Diplomarbeit, Fachhochschule Ingolstadt, Fachbereich Elektrotechnik und Informatik, 2008.
- [GKWZ07] GORBAN, ALEXANDER N., BALZS KGL, DONALD C. WUNSCH und ANDREI ZINOVYEV: *Principal Manifolds for Data Visualization and Dimension Reduction*. Springer Publishing Company, Incorporated, 2007.
- [Gmb07] GMBH, HEIM SYSTEMS: *DATARec 4 Series User Manual*. Friedrich-Ebert-Straße, TechnologiePark, D-51429 Bergisch Gladbach, Februar 2007. Publication number: 54995302.
- [Gol89] GOLDBERG, DAVID E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1 Auflage, January 1989.
- [Guo05] GUO, Y.: *Algorithmen zur On-Board-Diagnose von Fahrwerksschäden an Schienenfahrzeugen*. Doktorarbeit, TU Berlin, 2005.

- [Hal99] HALL, MARK A.: *Correlation-based Feature Selection for Machine Learning*. Doktorarbeit, The University of Waikato, Hamilton, NewZealand, 1999.
- [Hay94] HAYKIN, S.: *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, 1994.
- [HBV05] HUYSMANS, JOHAN, BART BAESENS und JAN VANTHIENEN: *A Comprehensive SOM-Based Scoring System*. In: *MLDM*, Seiten 80–89, 2005.
- [HEA05] *HEAD Application Note: Modulationsanalyse, HEAD acoustics*. HEAD acoustics, 2005.
- [HEA06] *HEAD Application Note: Psychoakustische Analysen in ArtemiS I, Lautheits- und Schärfe-Berechnung in Artemis, HEAD acoustics*. HEAD acoustics, 2006.
- [HEA07] *HEAD Application Note: Psychoakustische Analysen in ArtemiS II, Rauigkeit, Schwankungsstärke und Tonalität, HEAD acoustics*. HEAD acoustics, 2007.
- [HEA10] *HEAD Application Note: ArtemiS Multichannel Analysis Software - Description of the basic version, HEAD acoustics*, 2010.
- [HEB08] HOLMBOM, ANNIKA, TOMAS EKLUND und BARBRO BACK: *Customer Portfolio Analysis using the SOM*. In: HOLMBOM, ANNIKA, TOMAS EKLUND und BARBRO BACK (Herausgeber): *19th Australasian Conference on Information Systems, ACIS 2008*. 19th Australasian Conference on Information Systems, ACIS 2008, 2008.
- [HG01] HAESE, KARIN und GEOFFREY J. GOODHILL: *Auto-SOM: Recursive Parameter Estimation for Guidance of Self-Organizing Feature Maps*. *Neural Comput.*, 13(3):595–619, 2001.
- [HH06] HAMMER, BARBARA und ALEXANDER HASENFUSS: *Neural Computation - LVQ, Vorlesung 3+1*. Beispielcode LVQ, Jul. 2006.
- [HN06] HAMMER, BARBARA und NICOLAS NEUBAUER: *On the capacity of unsupervised recursive neural networks for symbol processing*. Technischer Bericht, Institute of Computer Science, Clausthal University of Technology, 2006.
- [HO08] HAJEK, PETR und VLADIMÍR OLEJ: *Air Quality Modelling by Kohonen's Self-organizing Feature Maps and LVQ*. In: *WSEAS TRANSACTIONS on ENVIRONMENT and DEVELOPMENT*, Band 4, 2008.
- [Hop82] HOPFIELD, J. J.: *Neural networks and physical systems with emergent collective computational abilities*. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, April 1982.
- [HS97] HALL, M. A. und L. A. SMITH: *Feature subset selection: a correlation based filter approach*. *International Conference on Neural Information Processing and Intelligent Information Systems.*, 0:855–858, 1997.
- [HS99] HALL, MARK A. und LLOYD A. SMITH: *Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper*. In: *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*, Seiten 235–239. AAAI Press, 1999.



- [Hu01] HU, YU HEN: *knndemo.m - demonstration of k-nearest neighbor classifier*, 2001.
- [HV02] HAMMER, BARBARA und THOMAS VILLMANN: *Generalized relevance learning vector quantization*. Neural Netw., 15(8-9):1059–1068, 2002.
- [Höv05] HÖVELMANN, K.: *Künstliche Intelligenz und künstliche neuro-nale Netze*. Technischer Bericht, Institut für Geodäsie, RWTH Aachen, 2005.
- [Jah01] JAHNKE, R.: *Beitrag zur akustischen Qualitätsprüfung im Bereich industrieller Fertigung*. Doktorarbeit, TU Berlin, 2001.
- [Jol86] JOLLIFFE, I.T.: *Principal Component Analysis*. Springer, New York, NY, 1986.
- [Kan94] KANGAS, JARI: *On the Analysis of Pattern Sequences by Self-Organizing Maps*. Doktorarbeit, Laboratory of Computer and Information Science, Helsinki University of Technology, Helsinki, Finland, 1994.
- [Kan04] KANGIESSER, TINO: *Selbstorganisierende Karten*. Technischer Bericht, Westfälische Wilhelms-Universität Münster, Institut für angewandte Informatik, 2004.
- [Ker08] KERN, MARCUS: *Ein Beitrag zur Erweiterung von Beamforming-Methoden*. Doktorarbeit, Technischen Universität Berlin, 2008.
- [Köh98] KÖHLER, JÜRGEN: *Merkmalsselektion zur Echtzeitdiagnose im Fahrzeug mittels neuronaler Netze*. Doktorarbeit, Universität Tübingen, 1998.
- [KJ97] KOHAVI, RON und GEORGE H. JOHN: *Wrappers for feature subset selection*. Artif. Intell., 97(1-2):273–324, 1997.
- [KK96] KANGAS, JARI und TEUVO KOHONEN: *Developments and Applications of the Self-Organizing Map and Related Algorithms*. Math Comput Simulat, 41:3–12, 1996.
- [KKL90] KANGAS, JARI, TEUVO KOHONEN und JORMA LAAKSONEN: *Variants of Self-Organizing Maps*. IEEE Transactions on Neural Networks, 1:93–99, 1990.
- [Koh01] KOHONEN, TEUVO: *Self-organizing maps*. Springer, 3 Auflage, 2001.
- [Kon94] KONONENKO, IGOR: *Estimating Attributes: Analysis and Extensions of RELIEF*. Seiten 171–182. Springer Verlag, 1994.
- [Kot08] KOTTE, M.: *Klassifikation von Fahrzeuggeräuschen zur Qualitätssicherung - Entwurf eines lernfähigen akustischen Klassifikators*. Diplomarbeit, Volkswagen AG und Technische Universität Ilmenau, 2008.
- [KR92] KIRA, KENJI und LARRY A. RENDELL: *The Feature Selection Problem: Traditional Methods and a New Algorithm*. In: AAAI, Seiten 129–134, Cambridge, MA, USA, 1992. AAAI Press and MIT Press.
- [KS95] KOHAVI, RON und DAN SOMMERFIELD: *Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology*, 1995.
- [Lam05] LAMPERT, JASMIN BRIGITTE: *Multivariate Statistische Analyse der Qualitätskontrollmerkmale von Affymetrix GeneChips*. Diplomarbeit, Fachhochschule Weihenstephan, 2005.

- [Lar05] LAROSE, DANIEL T.: *Discovering knowledge in data: an introduction to data mining*. John Wiley and Sons, 2005.
- [LCWE06] LAL, T. N., O. CHAPELLE, J. WESTON und A. ELISSEEFF: *Embedded methods*, Seiten 137–165. *Studies in Fuzziness and Soft Computing* ; 207. Springer, Berlin, Germany, 2006.
- [Le05] LE, VAN BANG: *Vorlesungsskriptum: Kohonen-Karten*. Universität Rostock, Fakultät für Informatik und Elektrotechnik, Institut für Informatik, 2005.
- [Leu02] LEUSTEAN, L.: *Liquid flow time series prediction using feed-forward neural networks and SuperSAB learning algorithm*. Technischer Bericht, National Institute for Research and Development in Informatics-ICI, 2002.
- [LKP06] LETTMANN, DR. THEO, OLIVER KRAMER und STEFFEN PRIESTERJAHN: *Vorlesungsskriptum: Selbstorganisierende Karten*. Universität Paderborn, FG Wissensbasierte Systeme, 2006.
- [LM98] LIU, HUAN und HIROSHI MOTODA: *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.
- [LMR98] LOUIS, A. K., P. MAASS und A. RIEDER: *Wavelets: Theorie und Anwendungen, Teubner Studienbücher*. Vieweg und Teubner Verlag, 1998.
- [Lou91] *Deutsche Norm DIN 45631: Berechnung des Lautstärkepegels und der Lautheit aus dem Geräuschspektrum Verfahren nach E. Zwicker*. DIN Deutsches Institut für Normung e.V., 1991.
- [Lou08] *Deutsche Norm DIN 45631/ A1 (Norm-Entwurf): Berechnung des Lautstärkepegels und der Lautheit aus dem Geräuschspektrum - Verfahren nach E. Zwicker - Änderung 1: Berechnung der Lautheit zeitvarianter Geräusche*. DIN Deutsches Institut für Normung e.V., 2008.
- [Mah36] MAHALANOBIS, P. C.: *On the generalised distance in statistics*. In: *Proceedings National Institute of Science, India*, Band 2, Seiten 49–55, April 1936.
- [MBGMF04] MLADENIĆ, DUNJA, JANEZ BRANK, MARKO GROBELNIK und NATASA MILIC-FRAYLING: *Feature selection using linear classifier weights: interaction with classification models*. In: *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, Seiten 234–241, New York, NY, USA, 2004. ACM.
- [MD96] MAIER, HOLGER R. und GRAEME C. DANDY: *The use of artificial neural networks for the prediction of water quality parameters*. *Water Resource Research*, 32(4):1013–1022, 1996.
- [Mel07] MELZER, THOMAS: *Statistische Mustererkennung WS 2007*. Internet, 2007.
- [Mey06] MEYER, MARTIN: *Signalverarbeitung - Analoge und digitale Signale, Systeme und Filter*. Vieweg und Teubner Verlag, 2006.

- [MGS05] MARINAI, SIMONE, MARCO GORI und GIOVANNI SODA: *Artificial Neural Networks for Document Analysis and Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27:23–35, 2005.
- [Mül04] MÜLLER, GERHARD: *Taschenbuch der technischen Akustik*. Springer Verlag, 2004.
- [MM01] MARTINEZ, WENDY und ANGEL MARTINEZ: *Computational Statistics Toolbox Version 1.0 26-Sep-2001*. Computational Statistics Handbook with MATLAB, Sep. 2001. CRC Press.
- [Moo03] MOORE, ANDREW W.: *Information Gain*. Carnegie Mellon University Pittsburgh, 2003.
- [MP43] MCCULLOCH, WARREN und WALTER PITTS: *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biology, 5(4):115–133, December 1943.
- [Mös07] MÖSER, MICHAEL: *Technische Akustik*. Springer Verlag, 2007.
- [Mös10] MÖSER, MICHAEL: *Messtechnik der Akustik*. Springer Verlag, 2010.
- [MSS02] MANJUNATH, B. S., PHILIPPE SALEMBIER und THOMAS SIKORA: *Introduction to MPEG-7: Multimedia Content Description Interface*, Seite 396. John Wiley & Sons, 2002. ISBN: 978-0-471-48678-7.
- [MST94] MICHIE, D., D.J. SPIEGELHALTER und C.C. TAYLOR: *Machine Learning, Neural and Statistical Classification*. 1994.
- [MUT<sup>+</sup>05] MÖRCHEN, F., A. ULTSCH, M. THIES, I. LÖHKEN, M. NÖCKER, C. STAMM, N. EFT-HYMIU und M. KÜMMERER: *MusicMiner: Visualizing timbre distances of music as topographical maps*. Technischer Bericht 47, Dept. of Mathematics and Computer Science, University of Marburg, Germany, 2005.
- [MW03] MÖLLER, BIRGIT und DENIS WILLIAMS: *Principal Component Analysis (PCA)*. Technischer Bericht, Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg, 2003.
- [Neu98] NEUWIRTH, STEFAN: *Implementierung von Kohonen-Netzen für den Neuro-Computer Synapse-1 und SUN Workstations*. Diplomarbeit, Westfälische Wilhelms-Universität Münster, 1998.
- [Nev06] NEVIADOMSKI, M.: *Neuroinformatische Analyse von Sprachsignalen zur Evaluierung einer Stottertherapie*. Diplomarbeit, Fachhochschule Schmalkalden, 2006.
- [Nie88] NIEMANN, H.: *Klassifikation von Mustern*. Springer Verlag, Berlin, Heidelberg, New York, Tokyo, 1988.
- [OBS07] OPFER, H., S. BORK und G. SEITZ: *Richtlautsprecher zur Dialogunterstützung im Fahrzeug*. In: *Fortschritte der Akustik, DAGA 2007, 33. Deutsche Jahrestagung für Akustik*, 2007.
- [OM09] OREND, BERND und INGO MEYER: *Schadensfrüherkennung mittels Körperschall*. MTZ, 5, 2009.
- [Opf09] OPFER, HOLGER: *Patentschrift: DE 102007051261 A1 Verfahren und Vorrichtung zur akustischen Beurteilung eines Kraftfahrzeuges*, 2009.

- [Pap08] PAPULA, LOTHAR: *Mathematik für Ingenieure und Naturwissenschaftler*. Vieweg+Teubner Verlag, 2008.
- [PC95] PEDRAZZI, P. und A.M. COLLA: *Simple feature extraction for handwritten character recognition*. Image Processing, International Conference on, 3:3320, 1995.
- [PK08] PFISTER, BEAT und TOBIAS KAUFMANN: *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer Verlag Berlin, 2008.
- [PLD05] PENG, HANCHUAN, FUHUI LONG und CHRIS DING: *Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy*. IEEE Trans. Pattern Anal. Mach. Intell., 27(8):1226–1238, 2005.
- [Pol05] POLYTEC, FA.: *PNA QuickCheck: DIAGNOSEPAKET FÜR DIE AKUSTISCHE GÜTE-PRÜFUNG*. Technischer Bericht, 2005.
- [Pra01] PRANTE, H.U.: *Modeling judgements of environmental sounds by means of artificial neural networks*. Doktorarbeit, Technische Universität Berlin, 2001.
- [Put09] PUTZ, SIEGFRIED: *Prasseln oder Rasseln?* Volkswagen Autogramm, 5, 2009.
- [Qui86] QUINLAN, J. R.: *Induction of Decision Trees*. Mach. Learn., 1(1):81–106, 1986.
- [Qui93] QUINLAN, J. ROSS: *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [Rab90] RABINER, LAWRENCE R.: *A tutorial on hidden Markov models and selected applications in speech recognition*. Readings in speech recognition, Seiten 267–296, 1990.
- [RBK96] ROWLEY, HENRY A., SHUMEET BALUJA und TAKEO KANADE: *Neural Network-Based Face Detection*. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, 0:203, 1996.
- [RHW88] RUMELHART, DAVID E., GEOFFREY E. HINTON und RONALD J. WILLIAMS: *Learning representations by back-propagating errors*. Seiten 696–699, 1988.
- [Rie94] RIEDMILLER, MARTIN: *Advanced supervised learning in multi-layer perceptrons - From backpropagation to adaptive learning algorithms*. Computer Standards & Interfaces, 16(3):265–278, 1994.
- [RJ93] RABINER, LAWRENCE und BIING-HWANG JUANG: *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
- [RKFM02] RITTHOFF, O., R. KLINKENBERG, S. FISCHER und I. MIERSWA: *A hybrid approach to feature selection and generation using an evolutionary algorithm*, 2002.
- [RN88] RODGERS, JOSEPH L. und ALAN W. NICEWANDER: *Thirteen Ways to Look at the Correlation Coefficient*. The American Statistician, 42(1):59–66, 1988.
- [Roh03] ROHDENBURG, THOMAS: *Klassifikation von Audio-Signalen*. Diplomarbeit, Universität Bremen, 2003.

- [Ros58] ROSENBLATT, FRANK: *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*. Psychological Review, 65(6):386–408, 1958.
- [RT91] REISS, MICHAEL und JOHN G. TAYLOR: *Storing temporal sequences*. Neural Netw., 4(6):773–787, 1991.
- [Ruf09] RUF, DANIEL: *Erweiterung einer Geräuschdatenbank und Entwurf eines effizienten und transparenten Mustererkennungsverfahrens zur Messdatenanalyse*. Diplomarbeit, Technische Universität Ilmenau, Fakultät für Elektrotechnik und Informationstechnik, 2009.
- [RvK03] ROBNIK-ŠIKONJA, MARKO und IGOR KONONENKO: *Theoretical and Empirical Analysis of ReliefF and RReliefF*. Mach. Learn., 53(1-2):23–69, 2003.
- [SB98] SUMPTER, NEIL und ANDREW J. BULPITT: *Learning Spatio-Temporal Patterns for Predicting Object Behaviour*. Image Vision and Computing, 18:2000, 1998.
- [Sch04] SCHÖNWANDT, THOMAS: *Filmszenenklassifizierung anhand der Tonspur mit Hilfe des MPEG-7 Standards*. Diplomarbeit, Rheinisch-Westfälische Technische Hochschule Aachen, 2004.
- [Sch06] SCHOLZ, M.: *Approaches to analyse and interpret biological profile data*. Doktorarbeit, University of Potsdam, Germany, 2006.
- [Sch07] SCHONERT, DR.-ING. T.: *Einsatz neuronaler Netze in der Ottomotoren-Vorentwicklung*. Technischer Bericht, Universität Hannover, 2007.
- [Sch09] *Deutsche Norm DIN 45692: Messtechnische Simulation der Hörempfindung Schärfe*. DIN Deutsches Institut für Normung e.V., 2009.
- [Scr06] SCRUCICA, LUCA: *Subset selection in dimension reduction methods*. Technischer Bericht, Università degli Studi di Perugia, 2006.
- [Sei98] SEIFFERT, UDO: *Wachsende mehrdimensionale selbstorganisierende Karten zur Analyse bewegter Szenen*. Doktorarbeit, University of Magdeburg, Magdeburg, Germany, JUL 1998.
- [Sei01] SEIFFERT, UDO: *Growing Multi-Dimensional Self-Organizing Maps for Motion Detection*. In: SEIFFERT, UDO und LAKHMI C. JAIN (Herausgeber): *Self-Organizing Neural Networks: Recent Advances and Applications*, Band 78 der Reihe *Studies in Fuzziness and Soft Computing*, Seiten 95–120, Heidelberg, Germany, 2001. Springer-Verlag.
- [Sei02] SEIFFERT, UDO: *Artificial Neural Networks on Massively Parallel Computer Hardware*. In: VERLEYSEN, MICHEL (Herausgeber): *Proceedings of the 10. European Symposium on Artificial Neural Networks ESANN 2002*, Seiten 319–330, Evere, Belgium, 2002. D-Side Publications.
- [Sen09] SENDLER, ULRICH: *Das PLM Kompendium - Referenzbuch des Produkt-Lebenszyklus-Managements*. Springer, 2009. ISBN 978-3-540-87897-1.

- [SG04] SICK, BERNHARD und CHRISTIAN GRUBER: *Vorlesungsskriptum: Soft-Computing*. Universität Passau, 2004.
- [Sha48] SHANNON, C. E.: *A mathematical theory of communication*. Bell system technical journal, 27, 1948.
- [SM95] SEIFFERT, UDO und BERND MICHAELIS: *Three-dimensional Self-Organizing Maps for Classification of Image Properties*. In: ANNES, Seiten 310–313, 1995.
- [SM97] SEIFFERT, UDO und BERND MICHAELIS: *Growing 3D-SOM's with 2D-Input Layer as a Classification Tool in a Motion Detection System*. In: *International Journal of Neural Systems (IJNS)*, Band 8, Seiten 81–89, 1997.
- [SM01] SEIFFERT, U. und B. MICHAELIS: *Multi-dimensional self-organizing maps on massively parallel hardware*. In: ALLINSON, NIGEL, HUIJUN YIN, LESLEY ALLINSON und JON SLACK (Herausgeber): *Advances in self-organising maps*, Seiten 160 – 166. Springer, 2001.
- [Smi02] SMITH, LINDSAY I.: *A tutorial on Principal Components Analysis*. February 2002.
- [Sot93] SOTTEK, R.: *Modelle zur Signalverarbeitung im menschlichen Gehör*. Doktorarbeit, RWTH Aachen, 1993.
- [Sot94] SOTTEK, R.: *Gehörgerechte Rauigkeitsberechnung*. In: *Fortschritte der Akustik, DAGA 1994, 20. Deutsche Jahrestagung für Akustik*, 1994.
- [SRD<sup>+</sup>04] SINDHWANI, VIKAS, SUBRATA RAKSHIT, DIPTI DEODHARE, DENIZ ERDOGMUS, JOSE PRINCIPE und PARTHA NIYOGI: *Feature Selection in MLPs and SVMs based on Maximum Output Information*. Technischer Bericht, Max Planck Institute of Biological Cybernetics, Tuebingen, 2004.
- [SS01] SCHOELKOPF, BERNHARD und ALEXANDER J. SMOLA: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
- [SSnH98] SCHUSCHEL, DIETRICH, LONGBOW APACHE SOFTWARE und CHUN NAN HSU: *A Weight Analysis-based Wrapper Approach to Neural Nets Feature Subset Selection*, 1998.
- [ST01] SCHUKAT-TALAMAZZINI, E.-G.: *Automatische Spracherkennung*. ISBN 3-528-15492-6, 2001.
- [Stö09] STÖCKER, HORST: *Taschenbuch mathematischer Formeln und moderner Verfahren*. Harri Deutsch Verlag, 2009.
- [Ste99] STEPPER, K.: *Ein Beitrag zur akustischen Güteprüfung von Komponenten der Kfz.- und Automatisierungstechnik*. Doktorarbeit, TU Berlin, 1999.
- [Ste04] STEENHAGEN, SHAWN: *The First Automotive Implementation of a Digital Voice Enhancement System*. DSP Resource Guide, 2004.
- [Ste05] STEPHAN, DANIEL: *Konzeption und Implementierung eines Clusteringmechanismus fuer ein Online Informationssystem*. Technischer Bericht, In Zusammenarbeit mit dem Institut für Seeverkehrswirtschaft und Logistik, 2005.

- [Ste06] STETTER, MARTIN: *Vorlesungsskriptum: Selbstorganisierende Merkmalskarten*. TU München, Chair for Foundations of Software Reliability and Theoretical Computer Science, 2006.
- [SWS98] SHANNON, CLAUDE E., WARREN WEAVER und SHANNON: *The Mathematical Theory of Communication*. University of Illinois Press, September 1998.
- [Ter98] TERHARDT, ERNST: *Akustische Kommunikation: Grundlagen mit Hörbeispielen*. Springer Verlag, 1998.
- [Tib94] TIBSHIRANI, ROBERT: *Regression Shrinkage and Selection Via the Lasso*. Journal of the Royal Statistical Society, Series B, 58:267–288, 1994.
- [TOB09] TWIEG, STEFAN, HOLGER OPFER und HELMUT BEIKIRCH: *Correlation based method for acoustic condition recognition*. In: *1st IEEE Students Conference Germany*, 2009.
- [TOB10] TWIEG, STEFAN, HOLGER OPFER und HELMUT BEIKIRCH: *Vehicle noise pattern recognition by Self-Organizing Maps*. Signal Processing: An International Journal (SPIJ), ISSN: 1985-2339, 3(6):180–190, January 2010.
- [Tol90] TOLLENAERE, TOM: *SuperSAB: fast adaptive back propagation with good scaling properties*. Neural Netw., 3(5):561–573, 1990.
- [Tou06] TOURETZKY, DAVE: *Artificial Neural Networks class slides*. Class web page, 2006.
- [TSS82] TERHARDT, E., G. STOLL und M. SEEWANN: *Algorithm for extraction of pitch and pitch salience from complex tonal signals*. In: *The Journal of the Acoustical Society of America*, Band 71, Seiten 679–688, 1982.
- [TZ10] TWIEG, S. und B. J. ZIMMERMANN: *Acoustic clustering for vehicle based sounds*. In: *Fortschritte der Akustik, DAGA 2010, 36. Deutsche Jahrestagung für Akustik*, 2010.
- [Vap95] VAPNIK, VLADIMIR N.: *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [Vap98] VAPNIK, VLADIMIR N.: *Statistical Learning Theory*. Wiley-Interscience, September 1998.
- [Ves97] VESANTO, JUHA: *Using the SOM and Local Models in Time-Series Prediction*. In: *Helsinki University of Technology*, Seiten 209–214, 1997.
- [Voe02] VOEGTLIN, T.: *Recursive self-organizing maps*, 2002.
- [wCJ07] CHEN, XUE WEN und JONG CHEOL JEONG: *Minimum Reference Set Based Feature Selection for Small Sample Classifications*. Technischer Bericht, Department of Electrical Engineering and Computer Science, University of Kansas, 2007.
- [wCZvA06] CHEN, XUE WEN, XIANGYAN ZENG und DEBORAH VAN ALPHEN: *Multi-class feature selection for texture classification*. Pattern Recogn. Lett., 27(14):1685–1691, 2006.
- [Wer08] WERNER, MARTIN: *Signale und Systeme: Lehr- und Arbeitsbuch mit Matlab-übungen und Lösungen*. Vieweg und Teubner Verlag, 2008.

- [WF05] WITTEN, IAN H. und EIBE FRANK: *Data mining : practical machine learning tools and techniques*. Elsevier, Morgan Kaufman, Amsterdam [u.a.], 2. ed. Auflage, 2005.
- [WHH<sup>+</sup>89] WAIBEL, A., T. HANAZAWA, G. HINTON, K. SHIKANO und K. J. LANG: *Phoneme recognition using time-delay neural networks*. Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, 37(3):328–339, 1989.
- [Wie02] WIEHR, STEPHAN: *Automatische Klassifikation semistrukturierter Daten*. Diplomarbeit, Universität des Saarlandes, Technische Fakultät I - Informatik Mathematik -, 2002.
- [Wir00] WIRTH, G.: *Ein Fehlerdiagnosesystem zur automatischen Klassifikation akustischer Signale*. Doktorarbeit, Uni Karlsruhe, 2000.
- [WK06] WEI, LI und EAMONN KEOGH: *Semi-supervised time series classification*. In: *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, Seiten 748–753, New York, NY, USA, 2006. ACM.
- [WM04] WANG, YUHANG und FILLIA MAKEDON: *Application of Relief-F Feature Filtering Algorithm to Selecting Informative Genes for Cancer Classification using Microarray Data (poster paper)*. In: *To appear in Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, Stanford, California, 2004.
- [Wor08] WORLICH, SILKE: *Entwicklung eines Messverfahrens zur Bestimmung von Vorderwagengeräuschquellen*. Diplomarbeit, Volkswagen AG und Fachhochschule Braunschweig /Wolfenbuettel, 2008.
- [Wür03] WÜRZNER, KAY-MICHAEL: *Textkategorisierung : Entscheidungsbäume*. Universität Bielefeld, Juni 2003.
- [Wue97] WUERTENBERGER, MICHAEL: *Modellgestützte Verfahren zur Überwachung des Fahrzustands eines PKW*. Doktorarbeit, VDI-Verlag, 1997.
- [YFH<sup>+</sup>08] YE, KAI, K. ANTON FEENSTRA, JAAP HERINGA, ADRIAAN P. IJZERMAN und ELENA MARCHIORI: *Multi-RELIEF: a method to recognize specificity determining residues from multiple sequence alignments using a Machine-Learning approach for feature weighting*. Bioinformatics, 24(1):18–25, 2008.
- [YHC06] YANG, HONGWU, DEZHI HUANG und LIANHONG CAI: *Perceptually Weighted Mel-Cepstrum Analysis of Speech Based on Psychoacoustic Model*. IEICE - Trans. Inf. Syst., E89-D(12):2998–3001, 2006.
- [YLG04] YU, LEI, HUAN LIU und ISABELLE GUYON: *Efficient feature selection via analysis of relevance and redundancy*. Journal of Machine Learning Research, 5:1205–1224, 2004.
- [YP97] YANG, YIMING und JAN O. PEDERSEN: *A Comparative Study on Feature Selection in Text Categorization*. In: *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, Seiten 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.



- [Yu203]      *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*, 2003.
- [Zen07]      ZENG, JINGTING: *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*. Technischer Bericht, Temple University Philadelphia, 2007.
- [Zim09]      ZIMMERMANN, BENJAMIN: *Entwicklung eines CSharp Tools für psychoakustische Analysen von fahrzeugbeschreibenden Messdaten zur Detektion von akustischen Auffälligkeiten im Fahrzeuginnenraum*. Diplomarbeit, TU Clausthal, 2009.
- [ZL07]      ZHAO, Z. und H. LIU: *Searching for Interacting Features*. In: *The 20th International Joint Conference on AI (IJCAI-07)*, 2007.
- [ZWZ07]      ZHANG, XUEYING, YIPING WANG und ZHEFENG ZHAO: *A Hybrid Speech Recognition Training Method for HMM Based on Genetic Algorithm and Baum Welch Algorithm*. Innovative Computing ,Information and Control, International Conference on, 0:572, 2007.
- [ZZ07]      ZHENG, H. und Y. ZHANG: *Feature selection for high dimensional data in astronomy*, 2007.



# Anlagenverzeichnis

## A.1 Merkmalreduktion und Fingerabdruckberechnung

### Abschnitt 4.1 Oktavspektrum: Mittenfrequenzen in Hz

16	31.5	63	125	250	500	1000	2000	4000	8000
----	------	----	-----	-----	-----	------	------	------	------

### Abschnitt 4.1 Terzspektrum: Mittenfrequenzen in Hz

12.5	16	20	25	31.5	40	50	63	80	100	125
160	200	250	315	400	500	630	800	1000	1250	1600
2000	2500	3150	4000	5000	6300	8000	10000	12500		

## Kapitel 5 PostgreSQL Datenbankablage der akustischen Messdaten im Überblick:

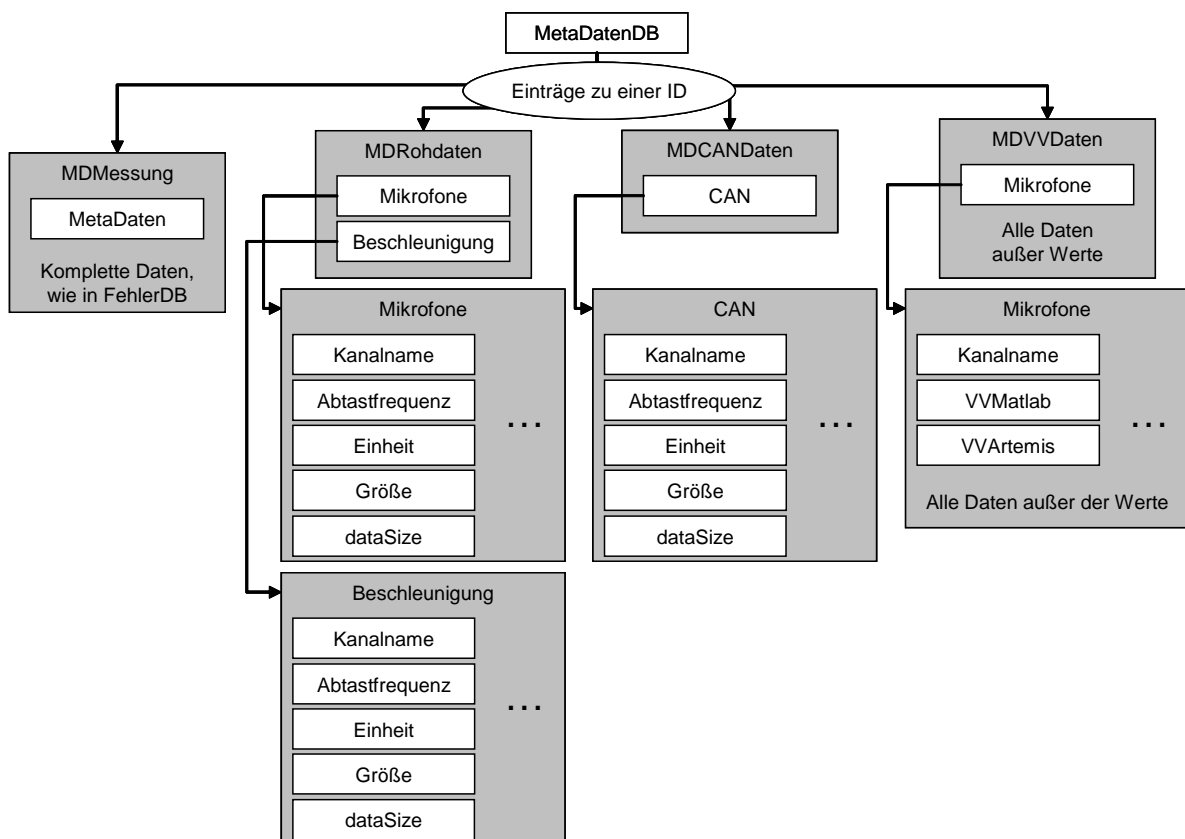


Abbildung A.1: Strukturierung der Datenablage in der entwickelten SQL-Datenbank.

### Abschnitt 5.1 Häufigkeitsverteilungen:

Abbildung A.2 zeigt die Häufigkeitsverteilungen der Merkmalsätze/ Muster gegenüber den Klassen für den Fehlerdatenbankdatensatz mit variabler Signalauschnittlänge  $t(n)$  (644 Merkmalsätze). Eine Aufteilung in Trainings- und Testmerkmalsätzen erfolgte mit dem Verhältnis 70% zu 30%. Fehlerklasse 12 hebt sich in der auftretenden Häufigkeit deutlich ab.

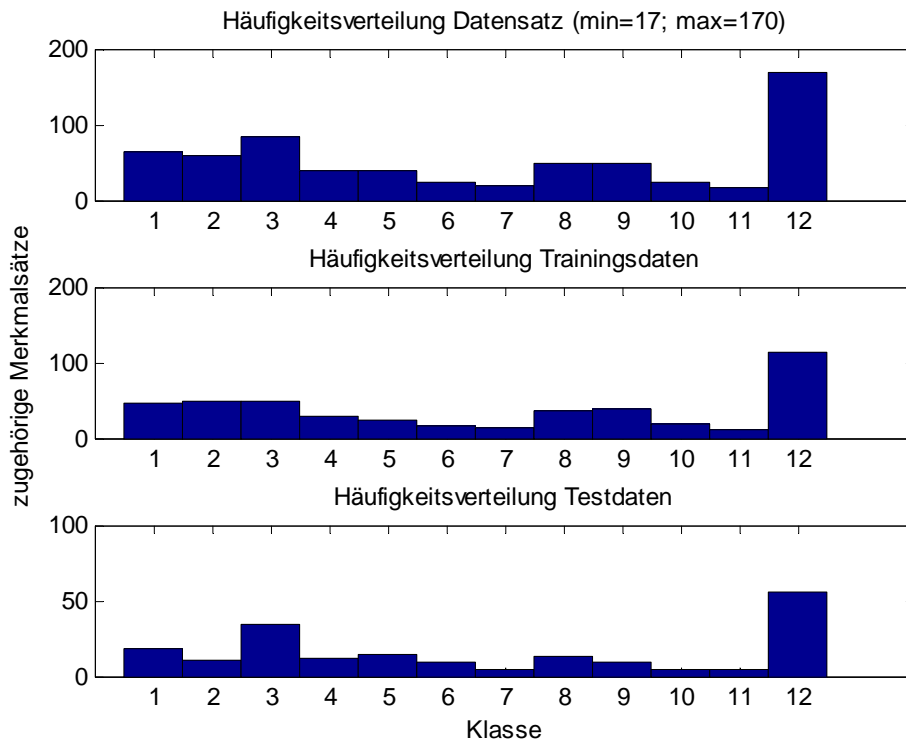


Abbildung A.2: Häufigkeitsverteilungen der Merkmalsätze/ Muster

Abbildung A.3 zeigt die Häufigkeitsverteilungen der Merkmalsätze/ Muster gegenüber den Klassen für den Fehlerdatenbankdatensatz mit variabler Signalauschnittlänge. Eine Aufteilung in Trainings- und Testmerkmalsätze erfolgte mit dem Verhältnis 70% zu 30%. Alle Fehlerklassen des Datensatzes sind jedoch gleichverteilt.

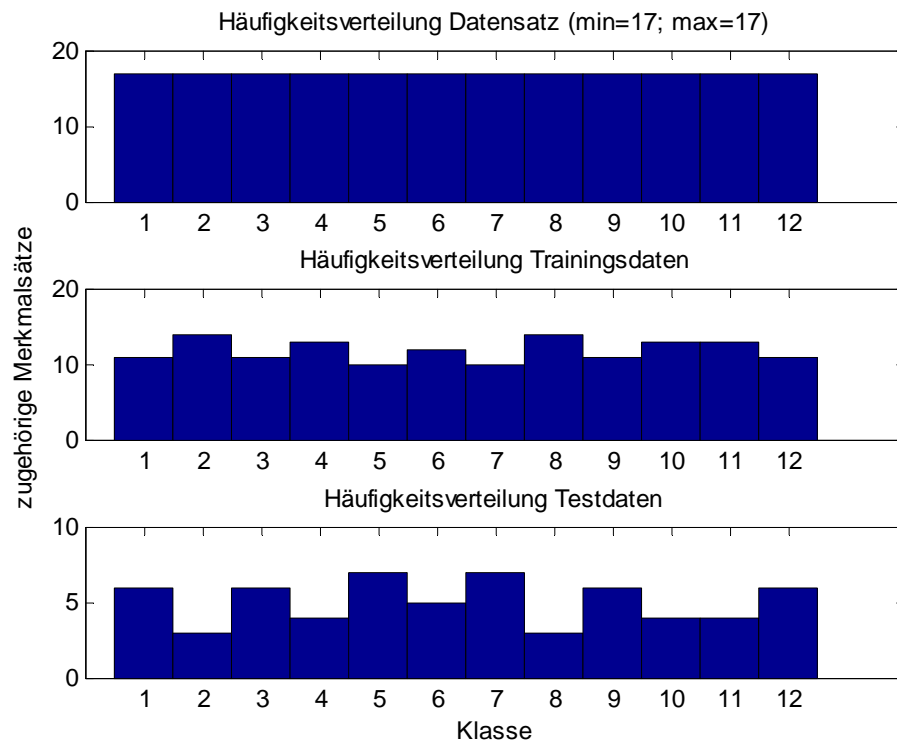


Abbildung A.3: Gleichverteilung der Merkmalsätze/ Muster

### Abschnitt 5.3.2 Fisher-Diskriminante:

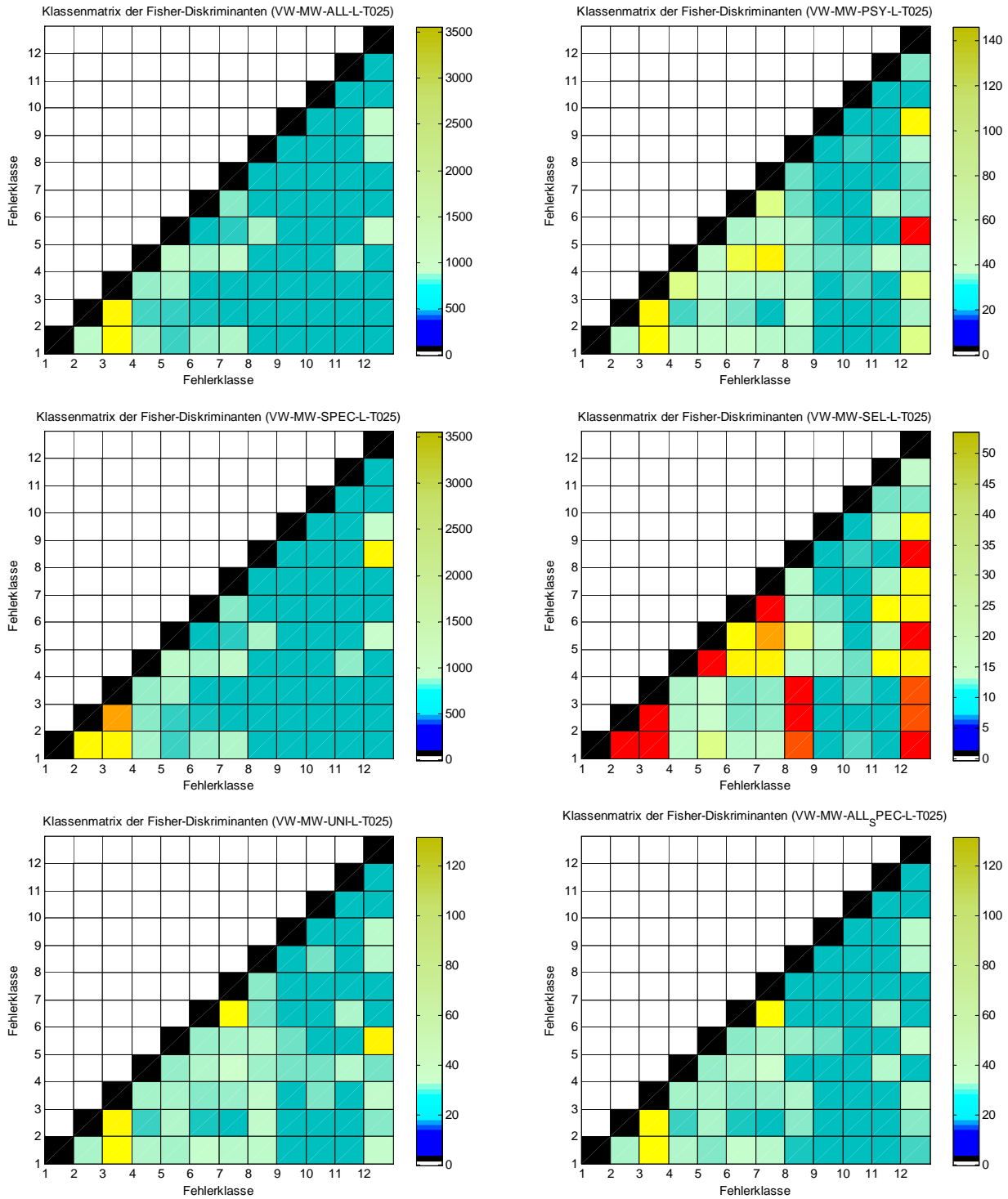


Abbildung A.4: Fisher-Diskriminanten Matrix für den Datensatz *ALL*, *PSY*, *SPEC*, *SEL*, *UNI* und *ALL-SPEC* bei  $t(n) = 250ms$  aus Abschnitt 5.3.2 (auf der Farbskala ist der FD-Wert  $f$  abgebildet)

a)	Ermittelte Fehlerklasse	Tatsächliche Fehlerklasse											
		1	2	3	4	5	6	7	8	9	10	11	12
		1	16	0	0	0	0	0	0	0	0	0	0
		2	0	22	0	0	0	0	0	0	0	0	0
		3	0	0	14	0	0	0	0	0	0	0	0
		4	0	0	0	17	0	0	0	0	0	0	0
		5	0	0	0	0	25	0	0	0	0	0	0
		6	0	0	0	0	0	19	0	0	0	0	0
		7	0	0	0	0	0	25	0	0	0	0	0
		8	0	0	0	0	0	0	23	0	0	0	0
		9	0	0	0	0	0	0	0	17	0	0	0
		10	0	0	0	0	0	0	0	0	20	0	0
		11	0	0	0	0	0	0	0	0	21	0	0
		12	0	0	0	0	0	0	0	0	0	26	0

b)	Ermittelte Fehlerklasse	Tatsächliche Fehlerklasse											
		1	2	3	4	5	6	7	8	9	10	11	12
		1	18	0	0	0	0	0	2	0	0	0	0
		2	0	25	0	0	0	0	0	0	0	0	0
		3	0	0	22	0	0	0	0	0	0	0	0
		4	0	0	0	16	0	0	0	1	0	0	0
		5	0	0	0	0	12	0	0	0	0	0	0
		6	0	0	0	0	0	20	0	0	0	0	0
		7	0	0	0	0	0	23	0	0	0	1	0
		8	0	0	0	0	0	0	23	0	0	0	0
		9	0	0	0	0	0	0	0	22	0	0	0
		10	0	0	0	0	0	0	0	0	20	0	0
		11	0	0	0	0	0	0	0	0	1	19	0
		12	0	0	0	0	0	0	0	0	0	0	20

c)	Ermittelte Fehlerklasse	Tatsächliche Fehlerklasse											
		1	2	3	4	5	6	7	8	9	10	11	12
		1	27	0	0	0	0	0	0	0	0	0	0
		2	0	19	1	0	0	0	0	0	0	0	0
		3	0	1	18	0	0	0	0	0	0	0	0
		4	0	0	0	22	0	0	0	0	0	0	0
		5	0	0	0	0	17	0	0	0	0	0	2
		6	0	0	0	0	0	17	0	0	0	0	0
		7	0	0	0	0	0	21	0	0	0	0	0
		8	0	0	0	0	0	0	13	0	0	0	0
		9	0	0	0	0	0	0	0	20	0	0	0
		10	0	0	0	0	0	0	0	0	20	0	0
		11	0	0	0	0	0	0	0	0	0	20	0
		12	0	0	0	0	0	0	0	0	0	0	27

d)	Ermittelte Fehlerklasse	Tatsächliche Fehlerklasse											
		1	2	3	4	5	6	7	8	9	10	11	12
		1	12	2	3	0	5	1	1	1	0	0	0
		2	1	6	5	0	1	0	0	1	0	0	0
		3	2	6	14	0	0	0	0	1	0	0	0
		4	0	1	1	18	2	1	0	0	0	0	0
		5	0	1	1	0	15	0	0	0	0	0	0
		6	0	0	0	0	0	18	6	0	0	0	0
		7	0	0	0	0	0	1	18	0	0	0	1
		8	0	0	0	0	0	0	0	20	0	0	1
		9	0	0	0	0	0	0	0	0	17	0	0
		10	0	0	0	0	0	0	0	0	0	19	4
		11	0	0	0	0	0	0	0	0	0	21	0
		12	0	0	0	0	1	0	0	0	1	1	14

e)	Ermittelte Fehlerklasse	Tatsächliche Fehlerklasse											
		1	2	3	4	5	6	7	8	9	10	11	12
		1	17	0	0	0	0	0	0	0	0	0	1
		2	2	20	0	0	0	0	0	0	0	0	0
		3	0	0	20	0	0	0	0	0	0	0	0
		4	0	0	0	26	0	0	0	0	0	0	0
		5	0	0	0	0	24	0	0	0	0	0	0
		6	0	0	0	0	0	19	1	0	0	0	0
		7	0	0	0	0	0	0	15	0	0	0	0
		8	0	0	0	0	0	0	0	20	0	0	0
		9	0	0	0	0	0	0	0	19	0	0	0
		10	0	0	0	0	0	0	0	0	18	0	0
		11	0	0	0	0	0	0	0	0	0	20	0
		12	0	0	0	0	0	0	2	0	0	0	21

f)	Ermittelte Fehlerklasse	Tatsächliche Fehlerklasse											
		1	2	3	4	5	6	7	8	9	10	11	12
		1	22	0	0	0	0	0	1	0	0	0	0
		2	0	19	0	0	0	0	0	0	0	0	0
		3	0	0	20	0	0	0	0	0	0	0	0
		4	0	0	0	16	0	0	0	0	0	0	0
		5	0	0	0	0	19	0	0	0	0	0	0
		6	0	0	0	0	0	22	2	0	0	0	2
		7	0	0	0	0	0	0	13	0	0	0	4
		8	0	0	0	0	0	0	0	20	0	0	0
		9	0	0	0	0	0	0	0	0	18	0	4
		10	0	0	0	0	0	0	0	0	0	19	0
		11	0	0	0	0	0	0	0	0	0	23	0
		12	0	0	0	0	0	0	0	0	1	0	20

Tabelle A.1: Wahrheitsmatrix für den Datensatz *ALL* (a), *PSY* (b), *SPEC* (c), *SEL* (d), *UNI* (e), *ALL-SPEC* (f) bei  $t(n) = 250ms$  aus Abschnitt 5.3.2

Tabelle A.2 zeigt die geordnete Darstellung der extrahierten Merkmale für die *FD*-Berechnung aus Abschnitt 5.3.2. Die Prozentzahl hinter dem Datensatznamen entspricht der ungefähren Klassifikationsgüte zur Trennung der 12 Fehlerklassen. Grün hervorgehoben sind die Merkmale, die in den verschiedenen Datensätzen mehrfach den maximalen *f*-Wert für eine Fehlerklassenkombination darstellen.

ALL	PSY	SPEC	SEL	UNI	ALL-SPEC
Spektrum Roughness (Zeitverlauf) [4]	Specific Loudness (Zeitverlauf) [32]	MFC [6]	Spectral Centroid	MFC [7]	Spectral Centroid
MFC [6]	Specific Loudness (Zeitverlauf) [212]	MFC [8]	Spectral Roll-off	MFC [11]	Spectral Roll-off
MFC [8]	Specific Loudness (Zeitverlauf) [227]	MFC [11]	Spectral Fluctuation	MFC [12]	Specific Loudness (Zeitverlauf) [4]
MFC [11]	Specific Loudness (Zeitverlauf) [239]	MFC [12]	Articulation Index (Zeitverlauf)	Spectral Centroid	Terzspektrum [10]
MFC [12]	Specific Loudness (Zeitverlauf) [240]	Cepstrum [4093]	Modulationsfrequenz (Zeitverlauf)	Specific Loudness [3]	Terzspektrum [13]
Cepstrum [4093]	Specific Loudness (Zeitverlauf) [4]	Spektrum gemittelt [1]	Loudness (Zeitverlauf)	Specific Loudness [4]	Terzspektrum [31]
Spektrum gemittelt [1]	Specific Fluctuation Strength [5]	Spektrum gemittelt [9]	Roughness (Zeitverlauf)	Specific Loudness [5]	Oktavspektrum [2]
Spektrum gemittelt [9]	Specific Fluctuation Strength [6]	Spektrum gemittelt [10]	Sharpness (Zeitverlauf)	Specific Loudness [6]	MFC [7]
Spektrum gemittelt [10]	Specific Fluctuation Strength [7]	Spektrum gemittelt [19]	Tonality (Zeitverlauf)	Specific Fluctuation Strength [5]	MFC [11]
Spektrum gemittelt [19]	Specific Fluctuation Strength [8]	Spektrum gemittelt [33]	Lenkgeschwindigkeit	Specific Fluctuation Strength [7]	MFC [12]
Spektrum gemittelt [33]	Specific Fluctuation Strength [9]	Spektrum gemittelt [58]	Lenkwinkel	Specific Fluctuation Strength [8]	Cepstrum [11]
Spektrum gemittelt [58]	Specific Fluctuation Strength [10]	Spektrum gemittelt [62]	Geschwindigkeit	Specific Fluctuation Strength [9]	Cepstrum [98]
Spektrum gemittelt [111]	Specific Fluctuation Strength [15]	Spektrum gemittelt [111]	Geschwindigkeit	Specific Fluctuation Strength [10]	Specific Fluctuation Strength [5]
Spektrum gemittelt [1100]	Specific Fluctuation Strength [16]	Spektrum gemittelt [1100]		Specific Fluctuation Strength [15]	Specific Fluctuation Strength [7]
Spektrum gemittelt [1461]	Specific Loudness Strength [21]	Spektrum gemittelt [1461]		Specific Fluctuation Strength [16]	Specific Fluctuation Strength [8]
Spektrum gemittelt [553]	Specific Loudness [35]	Spektrum gemittelt [1553]		Specific Fluctuation Strength [19]	Specific Fluctuation Strength [15]
Spektrum gemittelt [576]	Specific Loudness [60]	Spektrum gemittelt [1576]		Specific Loudness [18]	Specific Fluctuation Strength [16]
Spektrum gemittelt [1703]	Specific Loudness [229]	Spektrum gemittelt [1703]		Specific Loudness [31]	Specific Fluctuation Strength [19]
Spektrum gemittelt [1707]	Specific Loudness [239]	Spektrum gemittelt [1707]		Specific Loudness [187]	Specific Loudness [31]
Spektrum gemittelt [1747]	Specific Loudness [2]	Spektrum gemittelt [1716]		Specific Loudness [223]	Specific Loudness [239]
Spektrum gemittelt [1754]	Specific Loudness [3]	Spektrum gemittelt [1747]		Specific Loudness [228]	Specific Loudness [3]
Spektrum gemittelt [1802]	Specific Loudness [4]	Spektrum gemittelt [1754]		Specific Loudness [229]	Specific Loudness [4]
Spektrum gemittelt [1810]	Specific Loudness [5]	Spektrum gemittelt [1802]		Specific Loudness [239]	Specific Loudness [5]
Spektrum gemittelt [1918]	Specific Loudness [11]	Spektrum gemittelt [1810]		Specific Loudness [240]	Specific Loudness [5]
Spektrum gemittelt [1939]	Lenkgeschwindigkeit	Spektrum gemittelt [1918]		Spectral Roll-off	Lenkwinkel
Spektrum gemittelt [1960]	Lenkwinkel	Spektrum gemittelt [1939]		Lenkgeschwindigkeit	Geschwindigkeit
Spektrum gemittelt [1973]	Geschwindigkeit	Spektrum gemittelt [1960]		Lenkwinkel	
Spektrum gemittelt [1976]		Spektrum gemittelt [1973]		Geschwindigkeit	
Spektrum gemittelt [1988]		Spektrum gemittelt [1976]			
Spektrum gemittelt [2002]		Spektrum gemittelt [1981]			
Spektrum gemittelt [2011]		Spektrum gemittelt [1988]			
Spektrum gemittelt [2030]		Spektrum gemittelt [2002]			
Spektrum gemittelt [2035]		Spektrum gemittelt [2011]			
Spektrum gemittelt [2041]		Spektrum gemittelt [2030]			
Spektrum gemittelt [2042]		Spektrum gemittelt [2035]			
Spektrum gemittelt [2047]		Spektrum gemittelt [2041]			
Specific Fluctuation Strength [5]		Spektrum gemittelt [2042]			
Specific Fluctuation Strength [15]		Spektrum gemittelt [2047]			
Specific Loudness [4]		Lenkwinkel			
Lenkwinkel		Geschwindigkeit			
Geschwindigkeit					

Tabelle A.2: Geordnete Merkmale für die FD-Berechnung aus Abschnitt 5.3.2



### Abschnitt 5.3.3 Fisher-Diskriminante und Korrelationskoeffizient:

In Abbildung A.5 wird der extrahierte absolute Korrelationskoeffizient mit  $r < 0.3$  gezeigt. Je kleiner der Wert, desto geringer ist die Redundanz zwischen den Merkmalen für die jeweilige Kombination. Dargestellt ist die Matrix für den Datensatz PSY bei  $t(n) = 250ms$ . Auf der Farbskala ist der Absolutwert des Korrelationskoeffizienten  $r$  abgebildet.

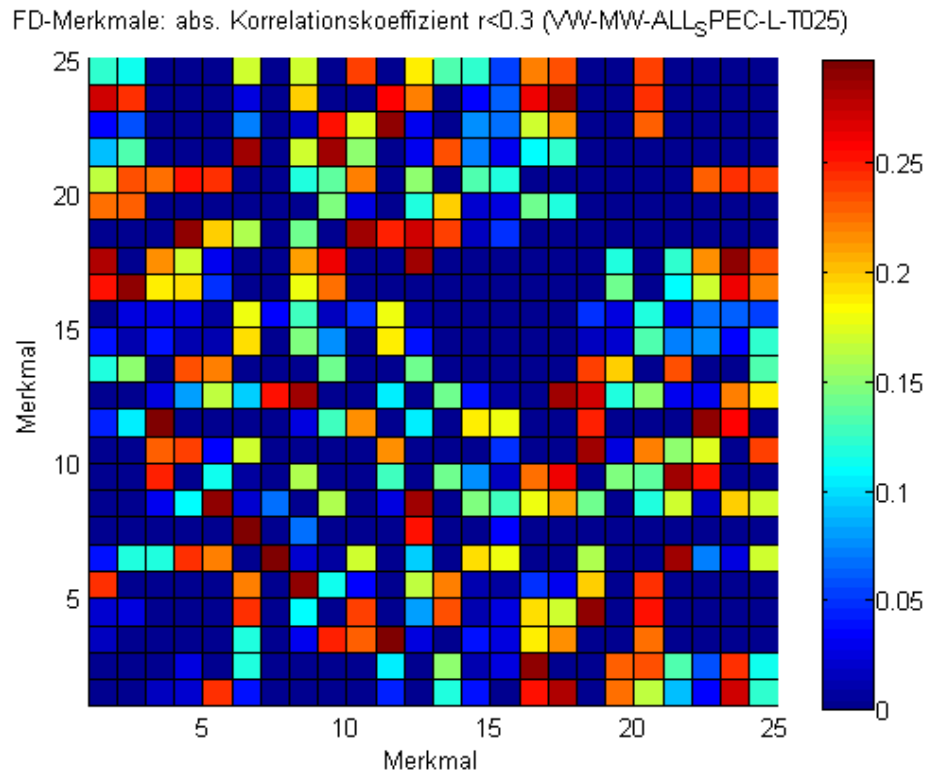


Abbildung A.5: Extrahierter absoluter Korrelationskoeffizient mit  $r < 0.3$

In Tabelle 5.11 sind die extrahierten Merkmale für die *FD*-Berechnung und anschließender Rangordnung durch das entwickelte Korrelationskoeffizientenverfahren gelistet. Die Prozentzahl hinter dem Datensatznamen entspricht der ungefähren Klassifikationsgüte zur Trennung der 12 Fehlerklassen bei 12 verwendeten Merkmalen. Grün hervorgehoben sind die Merkmale, die in den verschiedenen Datensätzen mehrfach den maximalen  $f$ -Wert für eine Fehlerklassenkombination darstellen (ohne „Spektrum gemittelt“).

ALL 93%	PSY 96%	SPEC 94%	SEL 79%	UNI 96%	ALL-SPEC 90%
Lenkwinkel	Specific Fluctuation Strength [8]	Lenkwinkel	Lenkwinkel	Specific Fluctuation Strength [8]	Specific Fluctuation Strength [8]
Giergeschwindigkeit	Specific Fluctuation Strength [6]	Giergeschwindigkeit	Spectral Fluctuation	Specific Fluctuation Strength [7]	Spectral Centroid
Spektrum gemittelt 1703	Specific Loudness [2]	Spektrum gemittelt 1703	Modulationsfrequenz (Zeitverlauf)	Specific Fluctuation Strength [10]	Specific Loudness [31]
Specific Roughness (Zeitverlauf) [4]	Specific Loudness [11]	MFC [12]	Spectral Centroid	Specific Fluctuation Strength [19]	Specific Roughness [4]
Spektrum gemittelt 1576	Specific Loudness [35]	Spektrum gemittelt 11	Geschwindigkeit	Lenkgeschwindigkeit	Cepstrum [98]
MFC [11]	Specific Fluctuation Strength [9]	Spektrum gemittelt 1576	Tonality (Zeitverlauf)	Specific Loudness [31]	Terzspektrum [10]
Spektrum gemittelt 1553	Specific Fluctuation Strength [10]	Spektrum gemittelt 2041	Articulationsindex (Zeitverlauf)	Specific Roughness [3]	Terzspektrum [31]
Spektrum gemittelt 11	Lenkgeschwindigkeit	Spektrum gemittelt 1960	Roughness (Zeitverlauf)	Spectral Centroid	MFC [7]
Spektrum gemittelt 1960	Specific Fluctuation Strength [21]	Spektrum gemittelt 1988	Lenkgeschwindigkeit	MFC [7]	MFC [12]
Spektrum gemittelt 1918	Lenkwinkel	Spektrum gemittelt 1981	Giergeschwindigkeit	Specific Roughness [6]	Specific Roughness [3]
Spektrum gemittelt 2041	Specific Loudness [4]	Spektrum gemittelt 2035	Sharpness (Zeitverlauf)	MFC [12]	Oktavspektrum [2]
Spektrum gemittelt 1988	Specific Fluctuation Strength [16]	Spektrum gemittelt 2047	Spectral Roll-off	Specific Fluctuation Strength [5]	Lenkwinkel
Spektrum gemittelt 2035	Specific Loudness (Zeitverlauf) [4]	Spektrum gemittelt 1976	Loudness (Zeitverlauf)	Specific Loudness [4]	Specific Fluctuation Strength [5]
Spektrum gemittelt 1976	Giergeschwindigkeit	Spektrum gemittelt 1553		Specific Loudness [187]	Giergeschwindigkeit
Spektrum gemittelt 2047	Specific Loudness [239]	Spektrum gemittelt 2002		Lenkwinkel	Specific Fluctuation Strength [19]
Spektrum gemittelt 2002	Specific Loudness (Zeitverlauf) [227]	Spektrum gemittelt 1918		Specific Loudness [240]	Specific Fluctuation Strength [15]
Spektrum gemittelt 1802	Specific Loudness (Zeitverlauf) [239]	Spektrum gemittelt 1802		Specific Fluctuation Strength [15]	Specific Loudness [5]
Spektrum gemittelt 1939	Specific Loudness [60]	Spektrum gemittelt 1939		Giergeschwindigkeit	Specific Loudness [239]
Spektrum gemittelt 1747	Specific Loudness (Zeitverlauf) [212]	Spektrum gemittelt 33		Specific Loudness [228]	Specific Fluctuation Strength [7]
Spektrum gemittelt 1707	Specific Fluctuation Strength [7]	MFC [8]		Specific Loudness [223]	Spectral Roll-off
Spektrum gemittelt 33	Specific Fluctuation Strength [5]	Spektrum gemittelt 9		Specific Fluctuation Strength [9]	Terzspektrum [13]
Spektrum gemittelt 1754	Specific Loudness [3]	Spektrum gemittelt 1707		Specific Loudness [18]	Specific Roughness (Zeitverlauf) [4]
Spektrum gemittelt 19	Specific Loudness (Zeitverlauf) [32]	Spektrum gemittelt 19		Spectral Roll-off	Cepstrum [1]
Spektrum gemittelt 10	Specific Loudness [5]	Spektrum gemittelt 1747		Specific Loudness [5]	MFC [11]
MFC [8]	Specific Fluctuation Strength [15]	Spektrum gemittelt 1100		Specific Loudness [239]	Specific Fluctuation Strength [16]
Specific Fluctuation Strength [15]	Specific Loudness [228]	Spektrum gemittelt 1754		Specific Fluctuation Strength [16]	
Spektrum gemittelt 58	Specific Loudness (Zeitverlauf) [240]	Spektrum gemittelt 58		Specific Loudness [229]	
Cepstrum [4093]		Cepstrum [4093]			
Specific Roughness [4]		Spektrum gemittelt 111			
Spektrum gemittelt 1461	MFC [11]				
MFC [12]	Spektrum gemittelt 1461				
Spektrum gemittelt 2042	Spektrum gemittelt 2042				
Spektrum gemittelt 2011	Spektrum gemittelt 2011				
Spektrum gemittelt 1973	Spektrum gemittelt 2030				
Spektrum gemittelt 2030	Spektrum gemittelt 1973				
Spektrum gemittelt 1810	Spektrum gemittelt 1810				
Spektrum gemittelt 9	MFC [6]				
MFC [6]	Spektrum gemittelt 10				
Specific Fluctuation Strength [5]	Spektrum gemittelt 62				
Spektrum gemittelt 111					

Tabelle A.3: Geordnete Merkmale für die *FD*-Berechnung und anschließender Rangordnung durch das Korrelationskoeffizientenverfahren

#### **Abschnitt 5.3.4 Fisher-Diskriminante und RELIEF:**

Tabelle A.4 zeigt die geordnete Darstellung der extrahierten Merkmale für die *FD*-Berechnung und anschließender Rangordnung durch das Mehrklassen-Relief-Verfahren. Die Prozentzahl hinter dem Datensatznamen entspricht der ungefähren Klassifikationsgüte zur Trennung der 12 Fehlerklassen bei 12 verwendeten Merkmalen. Grün hervorgehoben sind die Merkmale, die in den verschiedenen Datensätzen mehrfach den maximalen *f*-Wert für eine Fehlerklassenkombination darstellen (ohne „Spektrum gemittelt“).

ALL 99%	PSY 97%	SPEC 99%	SEL 80%	UNI 95%	ALL-SPEC 98%
Specific Fluctuation Strength [15]	Specific Fluctuation Strength [8]	Spektrum gemittelt 1707	Sharpness (Zeitverlauf)	Specific Fluctuation Strength [8]	Specific Fluctuation Strength [8]
Spektrum gemittelt 1703	Specific Fluctuation Strength [7]	Spektrum gemittelt 1703	Spectral Centroid	Specific Fluctuation Strength [10]	Specific Fluctuation Strength [7]
Spektrum gemittelt 1707	Specific Fluctuation Strength [10]	Spektrum gemittelt 1100	Geschwindigkeit	Specific Fluctuation Strength [7]	Specific Fluctuation Strength [15]
Spektrum gemittelt 1100	Specific Fluctuation Strength [9]	Spektrum gemittelt 19	Spectral Roll-off	Specific Fluctuation Strength [15]	Specific Fluctuation Strength [16]
Giergeschwindigkeit	Specific Fluctuation Strength [15]	Giergeschwindigkeit	Giergeschwindigkeit	Specific Fluctuation Strength [9]	Specific Fluctuation Strength [3]
Specific Fluctuation Strength [5]	Specific Fluctuation Strength [16]	MFC C [12]	Lenkwinkel	Specific Fluctuation Strength [16]	Specific Fluctuation Strength [5]
MFC C [8]	Specific Fluctuation Strength [5]	Spektrum gemittelt 58	Roughness (Zeitverlauf)	MFC C [11]	MFC C [7]
Spektrum gemittelt 19	Specific Fluctuation Strength [6]	Spektrum gemittelt 1	Loudness (Zeitverlauf)	Specific Fluctuation Strength [5]	MFC C [11]
MFC C [12]	Lenkwinkel	MFC C [8]	Lenkgeschwindigkeit	MFC C [7]	Specific Loudness [239]
Spektrum gemittelt 111	Specific Loudness [228]	Spektrum gemittelt 111	Articulationindex (Zeitverlauf)	Spectral Centroid	Giergeschwindigkeit
Spektrum gemittelt 1	Specific Loudness [239]	Lenkwinkel	Tonality (Zeitverlauf)	Giergeschwindigkeit	Specific Fluctuation Strength [19]
Spektrum gemittelt 58	Specific Fluctuation Strength [21]	Spektrum gemittelt 62	Spectral Fluctuation	Specific Roughness [4]	Spectral Centroid
MFC C [11]	Specific Roughness [11]	MFC C [11]	Modulationsfrequenz (Zeitverlauf)	Spectral Roll-off	Specific Roughness (Zeitverlauf) [4]
Lenkwinkel	Specific Roughness [4]	MFC C [6]		MFC C [12]	MFC C [12]
Specific Roughness (Zeitverlauf) [4]	Specific Roughness [3]	Cepstrum [4093]		Specific Roughness [3]	Specific Roughness [4]
Specific Roughness [4]	Specific Roughness [2]	Spektrum gemittelt 33		Specific Fluctuation Strength [19]	Spectral Roll-off
MFC C [6]	Specific Roughness [5]	Spektrum gemittelt 9		Specific Roughness [6]	Lenkwinkel
Spektrum gemittelt 33	Giergeschwindigkeit	Spektrum gemittelt 10		Specific Loudness [18]	Oktaspektrum [2]
Cepstrum [4093]	Specific Roughness (Zeitverlauf) [4]	Spektrum gemittelt 146		Specific Loudness [31]	Specific Loudness [31]
Spektrum gemittelt 9	Specific Loudness [60]	Spektrum gemittelt 153		Specific Loudness [187]	Specific Loudness [5]
Spektrum gemittelt 10	Specific Loudness [35]	Spektrum gemittelt 1461		Specific Loudness [11]	Cepstrum [11]
Spektrum gemittelt 153	Specific Loudness (Zeitverlauf) [32]	Spektrum gemittelt 1747		Specific Loudness [5]	Cepstrum [98]
Spektrum gemittelt 1461	Specific Loudness (Zeitverlauf) [212]	Spektrum gemittelt 1576		Specific Loudness [223]	Terzspektrum [10]
Spektrum gemittelt 1747	Specific Loudness (Zeitverlauf) [227]	Spektrum gemittelt 1754		Specific Loudness [229]	Terzspektrum [131]
Spektrum gemittelt 1576	Lenkgeschwindigkeit	Spektrum gemittelt 1802		Specific Loudness [228]	Terzspektrum [13]
Spektrum gemittelt 1754	Specific Loudness (Zeitverlauf) [239]	Spektrum gemittelt 1810		Specific Loudness [239]	
Spektrum gemittelt 1802	Specific Loudness (Zeitverlauf) [240]	Spektrum gemittelt 1918		Specific Loudness [240]	
Spektrum gemittelt 1810		Spektrum gemittelt 1939	Lenkgeschwindigkeit		
Spektrum gemittelt 1918		Spektrum gemittelt 1960			
Spektrum gemittelt 1939		Spektrum gemittelt 1976			
Spektrum gemittelt 1960		Spektrum gemittelt 1981			
Spektrum gemittelt 1976		Spektrum gemittelt 2002			
Spektrum gemittelt 1973		Spektrum gemittelt 2041			
Spektrum gemittelt 2002		Spektrum gemittelt 2011			
Spektrum gemittelt 2041		Spektrum gemittelt 1973			
Spektrum gemittelt 2011		Spektrum gemittelt 2035			
Spektrum gemittelt 2035		Spektrum gemittelt 1988			
Spektrum gemittelt 2047		Spektrum gemittelt 2030			
Spektrum gemittelt 1988		Spektrum gemittelt 2042			
Spektrum gemittelt 2030		Spektrum gemittelt 2047			
Spektrum gemittelt 2042					

Tabelle A.4: Geordnete Merkmale für die *FD*-Berechnung und anschließender Rangordnung durch das Mehrklassen-Relief-Verfahren

### **Abschnitt 5.3.5 Fisher-Diskriminante und Transinformation:**

Tabelle A.5 zeigt die geordnete Darstellung der extrahierten Merkmale für die *FD*-Berechnung und anschließender Rangordnung durch die Transinformation. Die Prozentzahl hinter dem Datensatznamen entspricht der ungefähren Klassifikationsgüte zur Trennung der 12 Fehlerklassen bei 12 verwendeten Merkmalen. Grün hervorgehoben sind die Merkmale, die in den verschiedenen Datensätzen mehrfach den maximalen *f*-Wert für eine Fehlerklassenkombination darstellen (ohne „Spektrum gemittelt“).

ALL 98%	PSY 98%	SPEC 98%	SEL 81%	UNI 96%	ALL-SPEC 96%
Spektrum gemittelt1707	Specific Roughness [4]	Spektrum gemittelt1707	Lenkwinkel	Specific Roughness [4]	Specific Roughness [4]
Spektrum gemittelt19	Giergeschwindigkeit	Spektrum gemittelt19	Spectral Roll-off	Giergeschwindigkeit	Giergeschwindigkeit
Spektrum gemittelt1100	Specific Fluctuation Strength [15]	Spektrum gemittelt1100	Geschwindigkeit	Specific Loudness [229]	Specific Loudness [239]
Spektrum gemittelt1703	Specific Loudness [239]	Spektrum gemittelt1703	Giergeschwindigkeit	MFC [11]	MFC [11]
MFC [11]	Specific Roughness [3]	MFC [11]	Spektral Centroid	Specific Roughness [3]	Specific Roughness [3]
Lenkwinkel	Spektrum gemittelt1746	Lenkwinkel	Lenkgeschwindigkeit	Spektral Centroid	Terzspektrum [31]
Specific Roughness [4]	Specific Fluctuation Strength [21]	Lenkwinkel	Roughness (Zeitverlauf)	Specific Loudness [5]	Spectral Centroid
Spektrum gemittelt1802	Specific Roughness [5]	Spektrum gemittelt58	Sharpness (Zeitverlauf)	Specific Roughness [5]	Specific Roughness [5]
MFC [8]	Specific Loudness (Zeitverlauf) [240]	MFC [8]	Tonality (Zeitverlauf)	Specific Loudness [239]	Lenkwinkel
Spektrum gemittelt1747	Specific Loudness [35]	Spektrum gemittelt1810	Loudness (Zeitverlauf)	Specific Fluctuation Strength [15]	Specific Fluctuation Strength [15]
Giergeschwindigkeit	Specific Fluctuation Strength [16]	Spektrum gemittelt1	Modulationsfrequenz (Zeitverlauf)	MFC [12]	Terzspektrum [13]
Spektrum gemittelt58	Lenkgeschwindigkeit	MFC [12]	Articulationsindex (Zeitverlauf)	Specific Roughness [6]	Spectral Roll-off
Spektrum gemittelt1461	Specific Fluctuation Strength [6]	Spektrum gemittelt1747	Spectral Fluctuation	Specific Loudness [223]	MFC [12]
MFC [12]	Specific Loudness [60]	Spektrum gemittelt33		Specific Loudness [18]	Cepstrum [1]
Spektrum gemittelt1976	Specific Loudness (Zeitverlauf) [212]	Giergeschwindigkeit		Specific Fluctuation Strength [16]	Specific Fluctuation Strength [16]
Spektrum gemittelt33	Specific Loudness (Zeitverlauf) [239]	Spektrum gemittelt1976	Sharpness (Zeitverlauf)	Specific Loudness [228]	Specific Loudness [31]
Spektrum gemittelt1810	Specific Loudness [2]	Spektrum gemittelt1461		MFC [17]	Cepstrum [98]
Cepstrum [4093]	Specific Roughness [2]	Cepstrum [4093]		Specific Fluctuation Strength [19]	MFC [7]
Spektrum gemittelt2041	Specific Roughness (Zeitverlauf) [4]	Spektrum gemittelt1973		Specific Loudness [31]	Oktafspektrum [2]
MFC [6]	Specific Fluctuation Strength [5]	MFC [6]		Specific Loudness [240]	Specific Fluctuation Strength [5]
Spektrum gemittelt1973	Specific Loudness (Zeitverlauf) [227]	Spektrum gemittelt2041		Specific Fluctuation Strength [5]	Terzspektrum [10]
Spektrum gemittelt1553	Specific Fluctuation Strength [10]	Spektrum gemittelt1553		Specific Loudness [187]	Specific Fluctuation Strength [7]
Specific Fluctuation Strength [15]	Specific Loudness [228]	Spektrum gemittelt1802		Specific Fluctuation Strength [9]	Specific Fluctuation Strength [8]
Spektrum gemittelt1754	Specific Fluctuation Strength [9]	Spektrum gemittelt111		Specific Fluctuation Strength [10]	
Spektrum gemittelt111	Specific Fluctuation Strength [7]	Spektrum gemittelt1754		Specific Fluctuation Strength [7]	
Spektrum gemittelt1939	Specific Fluctuation Strength [8]	Spektrum gemittelt2030		Specific Fluctuation Strength [8]	
Spektrum gemittelt2030		Spektrum gemittelt1939			
Spektrum gemittelt1576		Spektrum gemittelt101			
Spektrum gemittelt101		Spektrum gemittelt1988			
Spektrum gemittelt2011		Spektrum gemittelt1576			
Specific Roughness (Zeitverlauf) [4]		Spektrum gemittelt2011			
Spektrum gemittelt1988		Spektrum gemittelt2047			
Specific Fluctuation Strength [5]		Spektrum gemittelt9			
Spektrum gemittelt1918		Spektrum gemittelt1918			
Spektrum gemittelt2047		Spektrum gemittelt1981			
Spektrum gemittelt9		Spektrum gemittelt2035			
Spektrum gemittelt2035		Spektrum gemittelt1960			
Spektrum gemittelt1960		Spektrum gemittelt2042			
Spektrum gemittelt2042		Spektrum gemittelt2002			
Spektrum gemittelt2002					

Tabelle A.5: Geordnete Merkmale für die *FD*-Berechnung und anschließender Rangordnung durch die Transinformation

### **Abschnitt 5.3.6 Fisher-Diskriminante und Informationsgewinn:**

Tabelle A.6 zeigt die geordnete Darstellung der extrahierten Merkmale für die *FD*-Berechnung und anschließender Rangordnung durch das Informationsgewinn-Merkmalsextraktionsverfahren. Die Prozentzahl hinter dem Datensatznamen entspricht der ungefähren Klassifikationsgüte zur Trennung der 12 Fehlerklassen bei 12 verwendeten Merkmalen. Grün hervorgehoben sind die Merkmale, die in den verschiedenen Datensätzen mehrfach den maximalen *f*-Wert für eine Fehlerklassenkombination darstellen (ohne „Spektrum gemittelt“).

ALL 96%	PSY 97%	SPEC 97%	SEL 82%	UNI 93%	ALL-SPEC 82%
Specific Roughness (Zeitverlauf) [4]	Specific Loudness (Zeitverlauf) [240]	MFC C [6]	Articulationindex (Zeitverlauf)	MFC C [11]	Cepstrum [1]
MFC C [6]	Specific Loudness (Zeitverlauf) [32]	MFC C [11]	Roughness (Zeitverlauf)	MFC C [7]	Okavspektrum [2]
MFC C [11]	Specific Loudness [2]	MFC C [8]	Modulationsfrequenz (Zeitverlauf)	Specific Loudness [240]	MFC C [11]
MFC C [8]	Specific Loudness [239]	MFC C [12]	Loudness (Zeitverlauf)	Spectral Centroid	Terzspektrum [10]
MFC C [12]	Specific Loudness [35]	Spektrum gemittelt [11]	Spectral Centroid	MFC C [12]	MFC C [7]
Spektrum gemittelt [11]	Specific Fluctuation Strength [16]	Spektrum gemittelt [33]	Specific Fluctuation	Specific Fluctuation Strength [19]	Terzspektrum [31]
Spektrum gemittelt [33]	Specific Loudness (Zeitverlauf) [212]	Spektrum gemittelt [58]	Sharpness (Zeitverlauf)	Specific Loudness [223]	Spectral Roll-off
Spektrum gemittelt [19]	Specific Loudness [3]	Spektrum gemittelt [62]	Lenkwinkel	Specific Loudness [228]	Specific Roughness (Zeitverlauf) [4]
Spektrum gemittelt [58]	Specific Loudness [60]	Spektrum gemittelt [19]	Spectral Roll-off	Specific Loudness [18]	MFC C [12]
Spektrum gemittelt [10]	Specific Fluctuation Strength [21]	Spektrum gemittelt [10]	Giergeschwindigkeit	Specific Fluctuation Strength [16]	Spectral Centroid
Spektrum gemittelt [576]	Specific Loudness [4]	Spektrum gemittelt [1576]	Lenkgeschwindigkeit	Specific Loudness [3]	Specific Loudness [239]
Specific Fluctuation Strength [15]	Specific Loudness [228]	Spektrum gemittelt [1703]	Geschwindigkeit	Specific Loudness [31]	Specific Fluctuation Strength [19]
Spektrum gemittelt [1703]	Specific Loudness (Zeitverlauf) [227]	Cepstrum [4093]	Tonality (Zeitverlauf)	Specific Loudness [187]	Specific Fluctuation Strength [16]
Cepstrum [4093]	Specific Loudness (Zeitverlauf) [239]	Spektrum gemittelt [1707]		Specific Loudness [4]	Terzspektrum [13]
Spektrum gemittelt [100]	Specific Loudness (Zeitverlauf) [4]	Spektrum gemittelt [9]		Specific Loudness [229]	Specific Roughness [3]
Spektrum gemittelt [9]	Specific Fluctuation Strength [15]	Spektrum gemittelt [1100]		Specific Loudness [239]	Specific Roughness [4]
Spektrum gemittelt [1707]	Specific Fluctuation Strength [5]	Spektrum gemittelt [1746]		Specific Loudness [6]	Specific Loudness [31]
Spektrum gemittelt [1461]	Giergeschwindigkeit	Spektrum gemittelt [1]		Specific Fluctuation Strength [15]	Cepstrum [98]
Spektrum gemittelt [1]	Specific Fluctuation Strength [8]	Spektrum gemittelt [1553]		Specific Loudness [5]	Specific Fluctuation Strength [15]
Spektrum gemittelt [1553]	Specific Fluctuation Strength [10]	Spektrum gemittelt [1461]		Giergeschwindigkeit	Giergeschwindigkeit
Spektrum gemittelt [1802]	Lenkgeschwindigkeit	Spektrum gemittelt [1802]		Lenkgeschwindigkeit	Specific Loudness [5]
Spektrum gemittelt [1747]	Specific Fluctuation Strength [9]	Spektrum gemittelt [1747]		Specific Fluctuation Strength [8]	Specific Fluctuation Strength [8]
Spektrum gemittelt [1754]	Lenkwinkel	Spektrum gemittelt [1754]		Specific Fluctuation Strength [10]	Lenkwinkel
Spektrum gemittelt [2047]	Specific Loudness [11]	Giergeschwindigkeit		Specific Fluctuation Strength [9]	Specific Fluctuation Strength [5]
Giergeschwindigkeit	Specific Loudness [5]	Spektrum gemittelt [1810]		Spectral Roll-off	Specific Fluctuation Strength [7]
Spektrum gemittelt [1810]	Specific Fluctuation Strength [6]	Spektrum gemittelt [1918]		Lenkwinkel	
Spektrum gemittelt [1918]	Specific Fluctuation Strength [7]	Spektrum gemittelt [1939]		Specific Fluctuation Strength [5]	
Spektrum gemittelt [1918]	Specific Fluctuation Strength [7]	Spektrum gemittelt [1960]		Specific Fluctuation Strength [7]	
Specific Fluctuation Strength [5]		Spektrum gemittelt [1973]			
Spektrum gemittelt [1960]		Spektrum gemittelt [2047]			
Spektrum gemittelt [1939]		Spektrum gemittelt [2042]			
Spektrum gemittelt [1973]		Spektrum gemittelt [1988]			
Spektrum gemittelt [1988]		Spektrum gemittelt [1976]			
Spektrum gemittelt [2042]		Spektrum gemittelt [1981]			
Spektrum gemittelt [2042]		Spektrum gemittelt [2002]			
Spektrum gemittelt [1976]		Spektrum gemittelt [2035]			
Spektrum gemittelt [2035]		Spektrum gemittelt [2011]			
Specific Roughness [4]		Spektrum gemittelt [2030]			
Spektrum gemittelt [2030]		Spektrum gemittelt [2041]			
Spektrum gemittelt [2002]		Lenkwinkel			
Spektrum gemittelt [2041]					
Lenkwinkel					

Tabelle A.6: Geordnete Merkmale für die *FD*-Berechnung und anschließender Rangordnung durch den Informationsgewinn



### **Abschnitt 5.3.7 Fisher-Diskriminante und FCBF:**

Tabelle A.7 zeigt die geordnete Darstellung der extrahierten Merkmale für die *FD*-Berechnung und anschließender Rangordnung durch das FCBF-Merkmalextraktionsverfahren. Die Prozentzahl hinter dem Datensatznamen entspricht der ungefähren Klassifikationsgüte zur Trennung der 12 Fehlerklassen bei 12 verwendeten Merkmalen. Grün hervorgehoben sind die Merkmale, die in den verschiedenen Datensätzen mehrfach den maximalen *f*-Wert für eine Fehlerklassenkombination darstellen (ohne „Spektrum gemittelt“).

ALL 99%	PSY 99%	SPEC 96%	SEL 79%	UNI 99%	ALL-SPEC 98%
Spektrum gemittelt 19	Specific Fluctuation Strength [16]	Spektrum gemittelt 19	Geschwindigkeit	Specific Loudness [223]	Specific Fluctuation Strength [8]
Spektrum gemittelt 2041	Specific Loudness [239]	Spektrum gemittelt 2041	Lenkwinkel	Specific Loudness [239]	Specific Loudness [239]
Spektrum gemittelt 2035	Specific Roughness [5]	Spektrum gemittelt 2035	Tonalität (Zeitverlauf)	Specific Roughness [6]	Specific Roughness [5]
Spektrum gemittelt 1960	Specific Fluctuation Strength [8]	Spektrum gemittelt 1981	Loudness (Zeitverlauf)	Specific Roughness [5]	Specific Roughness [4]
Spektrum gemittelt 1939	Specific Fluctuation Strength [7]	Spektrum gemittelt 1960	Sharpness (Zeitverlauf)	Specific Loudness [31]	Specific Roughness [3]
Spektrum gemittelt 1703	Specific Fluctuation Strength [10]	Spektrum gemittelt 1621	Roughness (Zeitverlauf)	Specific Loudness [18]	Specific Loudness [31]
Spektrum gemittelt 2011	Specific Fluctuation Strength [9]	Spektrum gemittelt 1939	Articulationindex (Zeitverlauf)	Specific Fluctuation Strength [10]	Specific Fluctuation Strength [19]
Spektrum gemittelt 1918	Specific Fluctuation Strength [6]	Spektrum gemittelt 1703	Spectral Fluctuation	Specific Roughness [4]	Specific Fluctuation Strength [16]
Specific Roughness [4]	Specific Loudness [4]	Spektrum gemittelt 2011	Spectral Centroid	Specific Roughness [3]	Specific Fluctuation Strength [7]
Specific Fluctuation Strength [15]	Specific Loudness [228]	Spektrum gemittelt 1918	Giergeschwindigkeit	Specific Loudness [229]	Specific Fluctuation Strength [5]
Specific Fluctuation Strength [5]	Specific Loudness [35]	Spektrum gemittelt 2047	Spectral Roll-off	Specific Loudness [228]	Specific Fluctuation Strength [15]
Spektrum gemittelt 2047	Specific Fluctuation Strength [15]	Spektrum gemittelt 2042	Modulationsfrequenz (Zeitverlauf)	Specific Fluctuation Strength [19]	Spectral Centroid
Spektrum gemittelt 2042	Specific Fluctuation Strength [5]	Spektrum gemittelt 2030	Lenkgeschwindigkeit	Specific Fluctuation Strength [16]	Spectral Roll-off
Spektrum gemittelt 2030	Specific Loudness [2]	Spektrum gemittelt 1988		Specific Fluctuation Strength [9]	Specific Roughness (Zeitverlauf) [4]
Spektrum gemittelt 1988	Specific Loudness [60]	Spektrum gemittelt 1976		Specific Fluctuation Strength [7]	Terzspektrum [10]
Spektrum gemittelt 1976	Specific Fluctuation Strength [21]	Spektrum gemittelt 1973		Specific Fluctuation Strength [5]	Terzspektrum [13]
Spektrum gemittelt 1973	Specific Loudness [3]	Spektrum gemittelt 1810		Specific Loudness [187]	Terzspektrum [31]
Spektrum gemittelt 1810	Specific Loudness (Zeitverlauf) [32]	Spektrum gemittelt 1802		Specific Fluctuation Strength [15]	Oktavspektrum [2]
Spektrum gemittelt 1802	Specific Loudness (Zeitverlauf) [227]	Spektrum gemittelt 1754		Specific Fluctuation Strength [8]	MFC [11]
Spektrum gemittelt 1754	Specific Loudness (Zeitverlauf) [227]	Spektrum gemittelt 1716		MFC [11]	MFC [11]
Spektrum gemittelt 1576	Specific Loudness (Zeitverlauf) [239]	Spektrum gemittelt 1576		MFC [11]	MFC [11]
Spektrum gemittelt 1553	Specific Loudness (Zeitverlauf) [240]	Spektrum gemittelt 1461		MFC [12]	Cepstrum [11]
Spektrum gemittelt 1461	Specific Loudness (Zeitverlauf) [4]	Spektrum gemittelt 1100		Specific Loudness [240]	Cepstrum [98]
Spektrum gemittelt 1100	Specific Loudness [11]	Spektrum gemittelt 1100		Spectral Roll-off	Lenkwinkel
Spektrum gemittelt 111	Lenkgeschwindigkeit	Spektrum gemittelt 111		Lenkgeschwindigkeit	Giergeschwindigkeit
Spektrum gemittelt 58	Lenkwinkel	Spektrum gemittelt 58		Lenkwinkel	
Spektrum gemittelt 2002	Giergeschwindigkeit	Spektrum gemittelt 2002		Giergeschwindigkeit	
Spektrum gemittelt 1747		Spektrum gemittelt 1747			
Spektrum gemittelt 1707		Spektrum gemittelt 1707			
Spektrum gemittelt 10		Spektrum gemittelt 10			
Spektrum gemittelt 9		Spektrum gemittelt 9			
Spektrum gemittelt 11		Spektrum gemittelt 11			
Spektrum gemittelt 33		Spektrum gemittelt 33			
Specific Roughness (Zeitverlauf) [4]		MFC [6]			
MFC [6]		MFC [8]			
MFC [8]		MFC [11]			
MFC [11]		MFC [12]			
Cepstrum [12]		Cepstrum [4093]			
Lenkwinkel		Lenkwinkel			
Giergeschwindigkeit		Giergeschwindigkeit			

Tabelle A.7: Geordnete Merkmale für die FD-Berechnung und anschließender Rangordnung durch das FCBF-Merkmalextraktionsverfahren

## A.2 Akustikdiagnose-System

Technische Daten der Mikrofonkapseln die für den Aufbau des iMAD-Systems untersucht wurden.

DPA 4060:

- Übertragungsbereich: 20 Hz bis 20 kHz
- S/N ratio: 71 dB(A)
- Durchmesser: 5.4 mm; Länge: 12.7 mm
- Richtcharakteristik: Kugel

MEO-65PN

- Übertragungsbereich: 20 Hz bis 16 kHz
- S/N ratio: > 38 dB
- Durchmesser: 6.0 mm; Länge: 5.2 mm
- Richtcharakteristik: Kugel

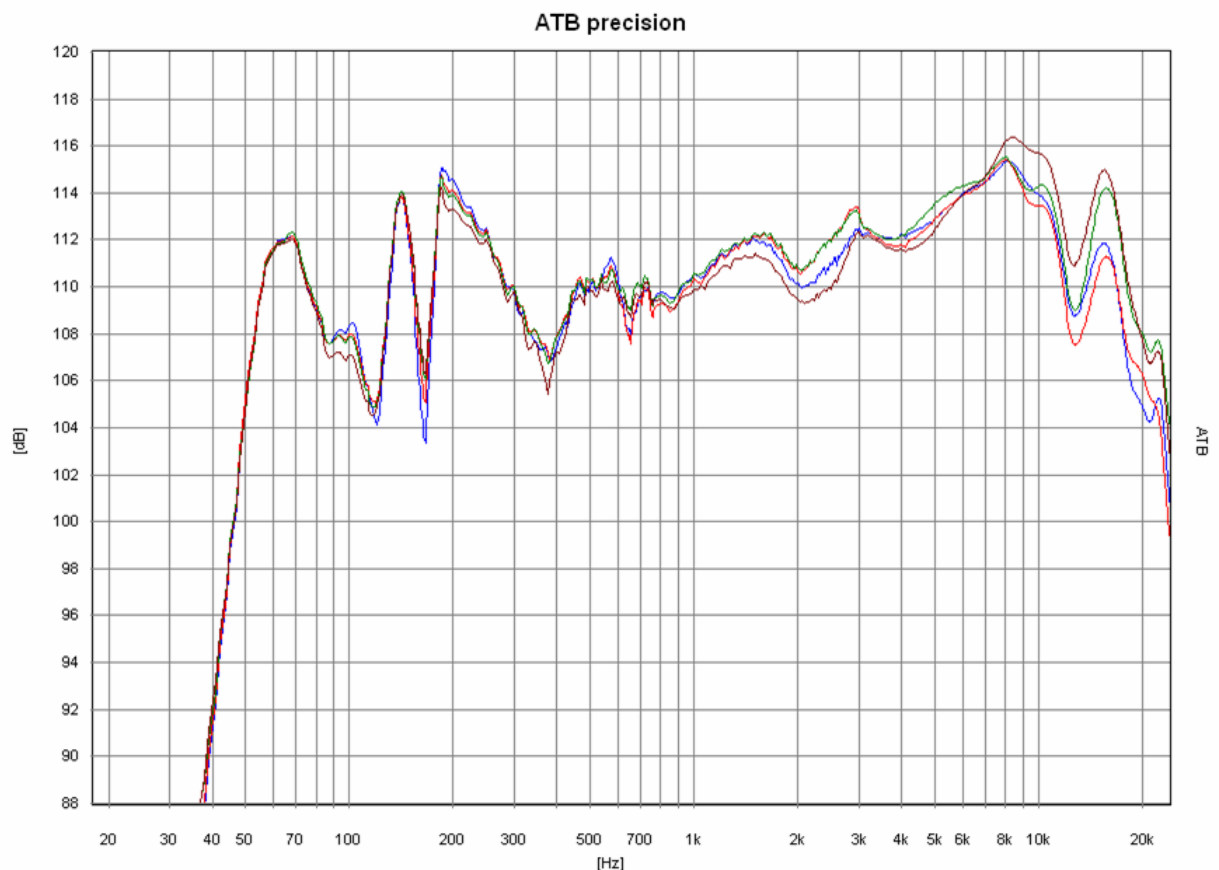


Abbildung A.6: Amplitudenfrequenzgang der Mikrofonkapsel des Typs DPA 4060. Es sind die Frequenzgänge von 4 Kapseln des gleichen Typs abgebildet.

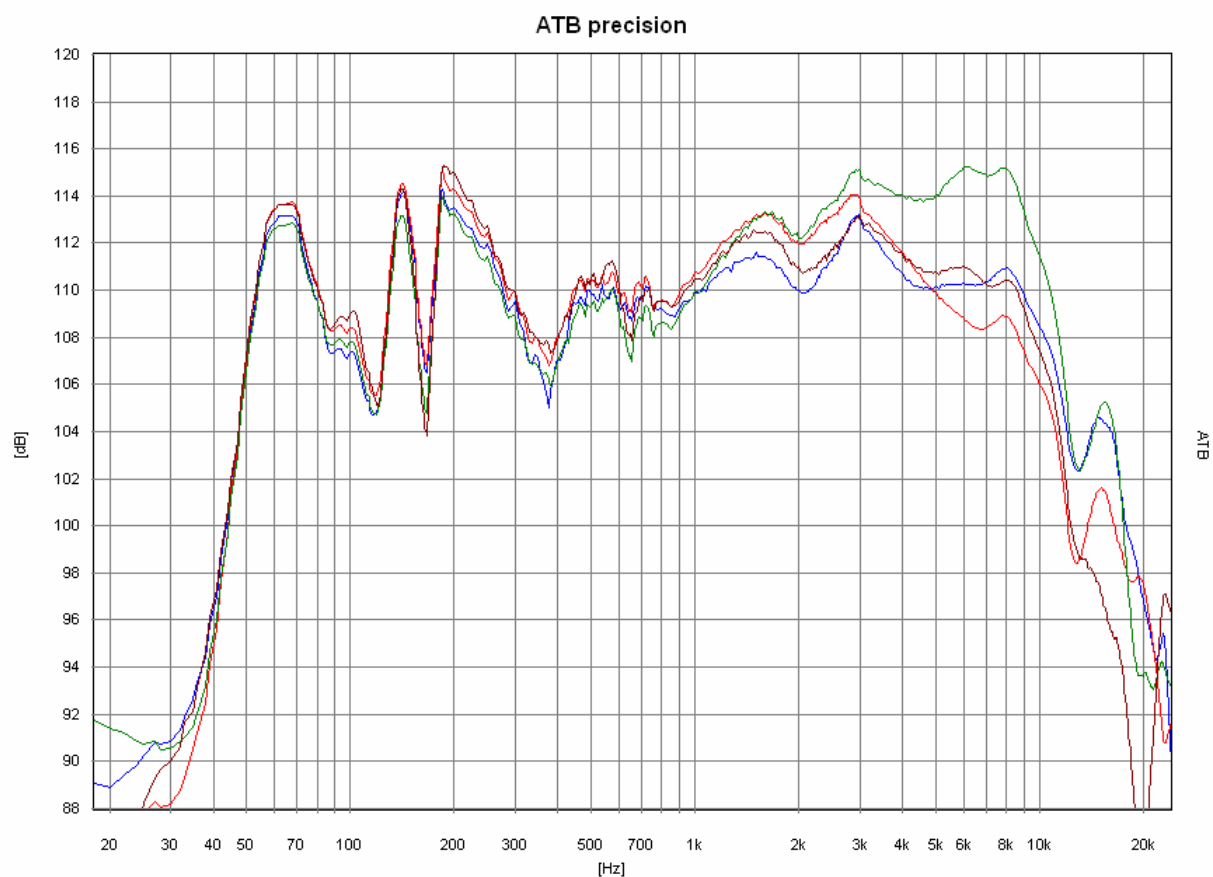


Abbildung A.7: Amplitudenfrequenzgang der Mikrofonkapsel des Typs MEO-65PN-02-623. Es sind die Frequenzgänge von 4 Kapseln des gleichen Typs abgebildet.

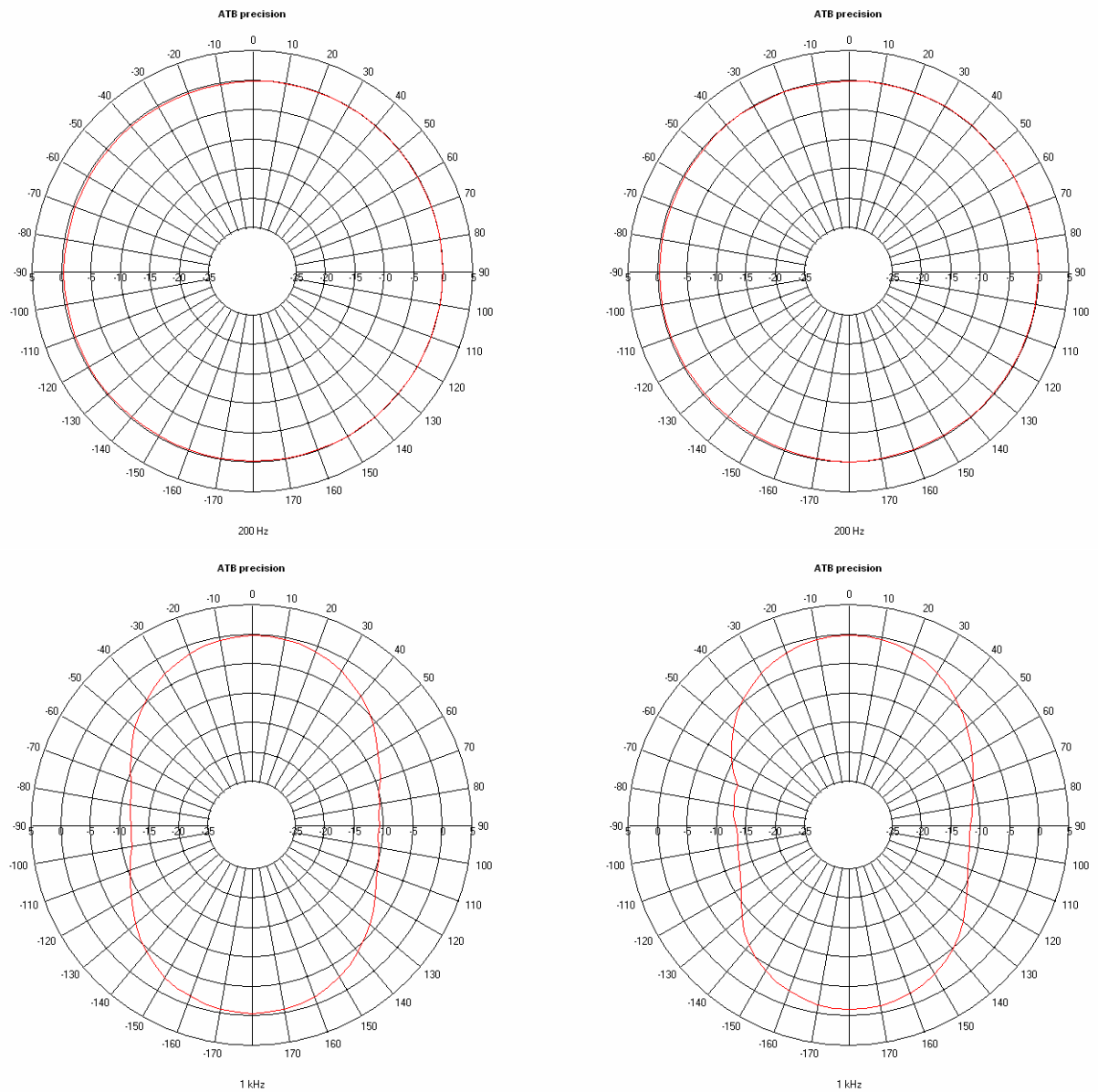


Abbildung A.8: Richtcharakteristik bei 0.2 kHz und 1 kHz für DPA 4060 (linke Seite) und MEO-65PN (rechte Seite)

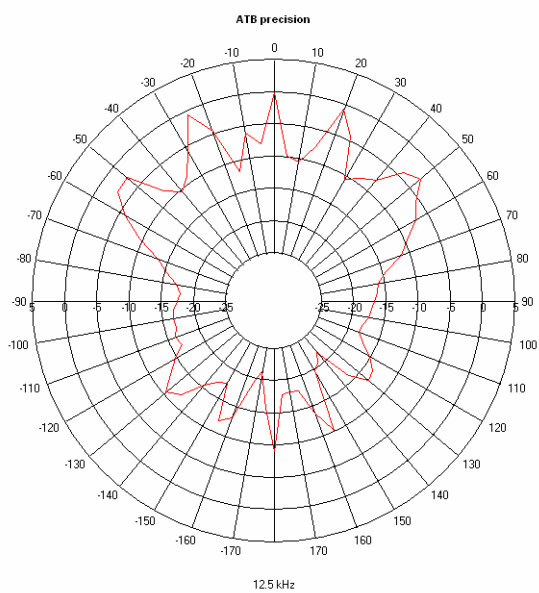
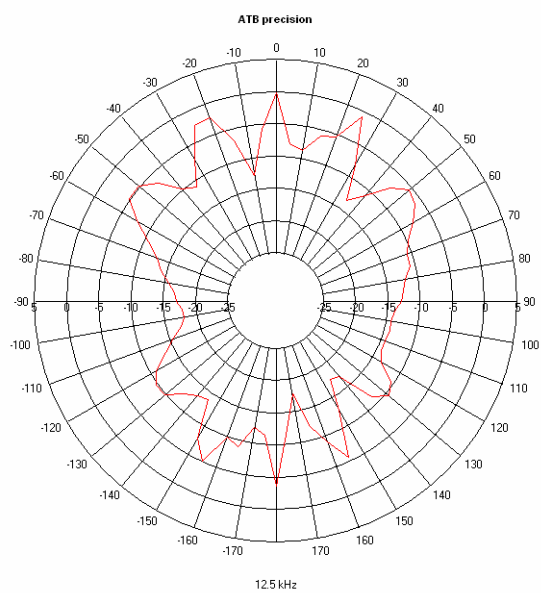
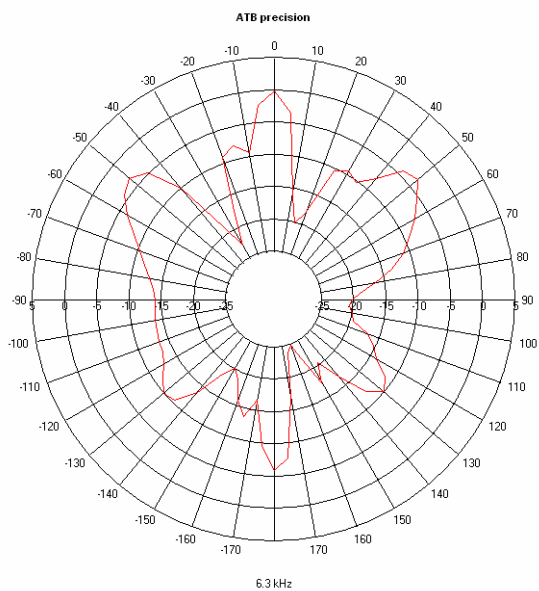
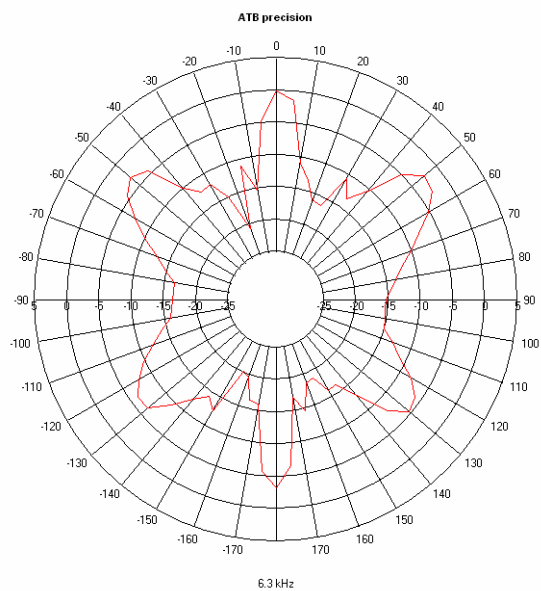


Abbildung A.9: Richtcharakteristik bei 6.3 kHz und 12.5 kHz für DPA 4060 (linke Seite) und MEO-65PN (rechte Seite)

# Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst habe, andere als die von mir angegebenen Quellen und Hilfsmittel nicht verwendet und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Stefan Twieg, Wolfsburg, 05.05.2010





# Thesen der Dissertation

1. Das subjektive akustische Empfinden im Fahrzeuginnenraum steht in direktem Zusammenhang mit dem Komfortempfinden und der wahrgenommenen Qualität des Fahrzeugs. Etwaige akustisch relevante Abweichungen im Produktionsprozess und Kundenbeanstandungen, bei denen ein hervortretendes Schallereignis auf ein verschlissenes oder defektes Bauteil hinweist, können derzeit nicht durch automatisierte Verfahren analysiert und beurteilt werden.
2. Durch ein Mikrofon lässt sich das Innengeräusch erfassen, über eine Signalverarbeitung lässt sich das Signal einer Mustererkennung zuführen. Eine Überwachung und selbstständige Analyse des Innenraumgeräuschs auch während der Nutzungsphase durch den Kunden (alternativ zum Produktionsprozess) ist prinzipiell realisierbar.
3. Die Dokumentation und Sicherung von akustischen Beanstandungen in Datenbanken ermöglicht eine Überwachung und selbstständige Analyse des Innenraumgeräuschs auch während der Nutzungsphase durch den Kunden.
4. Grundlage für die Algorithmenentwicklung bildet die automatisierte Berechnung zustandsbeschreibender Kenngrößen aus den Zeitsignalen. Dabei ist eine Beschreibung objektiver akustischer Messgrößen mit dem subjektiven Geräuschempfinden über psychoakustische Parameter möglich. Die Ableitung eines akustischen Fingerabdrucks aus den berechneten akustikbeschreibenden Kenngrößen ermöglicht eine Erkennung und Klassifikation von Fehlergeräuschen.
5. Verfahren zur Reduktion der Dimensionen wirken dem Zielkonflikt zwischen dem Berechnungsaufwand für eine große Menge an Merkmalen und einer möglichst hohen Erkennungsrate des Mustererkenners entgegen.
6. Verfahren zur Rangberechnung wie Transinformation, *Multi-Relief* und *Fast Correlation Based Filter* erreichen die höchsten Klassifikationsraten. Bereits bei neun durch den *Multi-Relief* ermittelten Merkmalen können Erkennungsrate von über 90 % für die Trennung von 12 verschiedenen Geräuscharten erreicht werden.
7. Der Fahrzeugtyp hat einen entscheidenden Einfluss auf die Merkmalausprägung. Ein universeller akustischer Fingerabdruck erreicht jedoch ähnlich hohe Erkennungsrate wie ein fahrzeugspezifischer Fingerabdruck. Eine Umsetzung der Berechnungsalgorithmen ist nach dieser Erkenntnis für zukünftige Anwendungen im Fahrzeug für alle Fahrzeugtypen übertragbar, es erfolgt lediglich eine Adaption der Wertebereiche.
8. Fahrzeugunabhängige Merkmale, die nicht durch den Fahrzeugtyp beeinflusst werden, sind bisher nicht identifiziert worden. Die Merkmale werden zeitinvariant durch den Klassifikator bewertet, da deren Berechnung gemittelt über das Zeitfenster  $t(n)$  durchgeführt wird. Der

charakteristische Zeitverlauf eines Geräuschs wird dabei vernachlässigt, dieser stellt jedoch ebenfalls eine Informationsquelle dar, die zukünftig als Merkmal genutzt werden könnte. So ist es einem geschulten menschlichen Gehör möglich, ein defektes Radlager unabhängig vom Fahrzeugtyp nicht zuletzt anhand der zeitlichen Änderung des Geräuschs zu erkennen.

9. Eine kleine Zeitkonstante  $t(n)$  bewirkt für die Berechnung der Merkmale eine genauere akustische Abbildung der Fehlerklassen und somit ein besseres Klassifikationsergebnis.
10. Selbstorganisierende Karten (SOM, neuronaler Algorithmus) interpretieren die berechneten akustischen Kenngrößen und unterscheiden die Fehlerzustände. Anhand des sogenannten SOM-Fehleragenten kann eine automatische Erkennung von defekten Geräuschen erfolgen. Parameteruntersuchungen des Modells zeigen eindeutige Abhängigkeiten von der Anzahl der verwendeten Neuronen.
11. Mehrwertige Merkmale, wie beispielsweise das Spektrum über das Zeitfenster  $t(n)$ , benötigen einen deutlich höheren Rechenaufwand als der einwertige Effektivwert der Messgröße. Der SOM-Fehleragent wird deshalb mit möglichst wenigen und einfach zu berechnenden einwertigen Merkmalen betrieben. Der Nachteil einer fahrzeugabhängigen Erweiterung der Lerndatenmenge wird deshalb durch Recheneffizienz kompensiert.
12. Für den Fall guter Hörbarkeit durch den Menschen kann das von den Fehlerbauteilen verursachte auffällige Innenraumgeräusch zuverlässig durch den SOM-Fehleragenten dem fehlerbehafteten Zustand zugeordnet werden. Bei schlechter Hörbarkeit, bei der eine eindeutige Wahrnehmung durch den Menschen kaum möglich ist, funktioniert der Fehleragent für fast alle Fehlerarten zuverlässig, jedoch mit geringerer Erkennungsrate.
13. Die nächste Evolutionsstufe des existierenden Geräuschlexikons wird durch das *integrierte Mikrofonarray-Diagnose-System* (iMAD-System) zur Geräuschaufnahme erreicht.
14. Einen erhöhten Bedarf an akustischer Diagnose von Fehlerzuständen lässt sich sowohl für die Fahrzeugentwicklung, als auch für den gesamten Bereich der Qualitätssicherung identifizieren. Aufgrund der im Rahmen dieser Arbeit genannten Ergebnisse schließen sich hier weitere Folgearbeiten an, die eine Bearbeitung der Geräuscherfassung am Kundenfahrzeug, deren Ablage, Verarbeitung und zukünftige Diagnosefunktionen (beispielsweise für das Soundsystem) ermöglichen.





# Danksagung

Diese Arbeit wurde im Rahmen einer Anstellung als Doktorand bei der Volkswagen AG erstellt. Das übergreifende Thema der *intelligenten akustischen Diagnose* wurde in der Konzernforschung durch Herrn Dr. rer. nat. Holger Opfer initiiert und von mir weitergeführt. Für eine unkomplizierte Zusammenarbeit, stets konstruktiven Anmerkungen und der souveränen Unterstützung bei der Bewältigung von VW-Prozessen danke ich meinem Unterabteilungsleiter Herrn Dr.-Ing. Jens Meschke. Aufbauend auf Voruntersuchungen von Martin Kotte, ersten Messfahrten von Silke Worlich und deren Weiterführung von Tim Berthold und Daniel Ruf konnte ich die Themenpunkte dieser Promotion bearbeiten. Dabei unterstützte auch stets die Forschungswerkstatt und insbesondere Klaus Kowalewicz beim Umbau der Versuchsfahrzeuge. Für die Unterstützung meiner Kollegen Markus Neubauer, Lukasz Janczweski, Dr. Gunnar Gäbel und Dr. Marcus Kern durch kritische Fragen sowie aufheiternde Kommentare möchte ich mich ebenfalls bedanken.

Die wissenschaftliche Betreuung erfolgte zum einen durch Herrn Prof. Dr.-Ing. habil. Helmut Beikirch, dem ich für seine stete Unterstützung beim Klären von Problemen und offenen Fragen danke. Zum anderen erfolgte die Betreuung durch Herrn Dr. Holger Opfer, der trotz seines Aufenthalts in Japan entweder in Telefonaten und Emails oder während kurzen Deutschland-Besuchen immer ein offenes Ohr und konstruktive fachübergreifende Tipps, Anmerkungen und Kritik für aktuelle Fragen hatte. Zudem möchte ich mich bei Herrn Prof. Dr.-Ing. Matthias Sturm bedanken, der bereits meine Diplomarbeit an der HTWK betreute und mein Promotionsvorhaben mit der Universität Rostock unterstützt hat.

Ganz besonders möchte ich an dieser Stelle auch meiner Lebensgefährtin Judith danken, die mir auch in familiär sehr schwierigen Zeiten aufbauend zur Seite stand.



# Index

- acc*, 19
- AI*, 25
- akustischer Fingerabdruck, 27, 84
- ANN, 42
- artificial neural network*, 42
- Artikulationsindex, 25
  
- Backpropagation Algorithmus, 44
- Bayes-Klassifikator, 56
- Beamforming*, 3, 126
- Bedingte Wahrscheinlichkeit, 33
- Blinde Quellentrennung, 3
- bootstrap, 40
- BSS, 3
  
- C4.5, 30
- Canberra-Metrik, 42
- Cart, 30
- Cepstrum, 24
- Classification Tree*, 65
- Cluster*, 41
- CM, 5
- component planes*, 96
- Controller Area Network, 60
- cross-validation*, 39
- CWT, 10
  
- Datensätze, 59
- dB, 26
- DCT, 24
- Decision Tree*, 54
- Delay-Line*, 53
- Delta-Regel, 44
- DFT, 10, 23
- Diagnosetester, 125
- Diskrete Kosinus-Transformation, 24
- Diskrete Wavelet-Transformation, 25
- Distanzmaße, 40
- DLL, 129
- DVE, 4
- DWT, 10, 25
- Dynamic Link Library*, 129
  
- Effektivwert, 6, 24
- Embedded-Methoden*, 29
- Entropie (Shannon), 33
- Entscheidungsbäume, 54
- Erkennungsrate, 8
- ETA, 5
- Euklidische Distanz, 41
  
- F-Ratio, 31
- Fast Correlation Based Filter, 35
- Fast-Correlation-Based-Filtering (FCBF), 64
- FD, 13, 31, 67
- Fehlerdatenbank, 59
- Fehlerindikation, 87
- FFT, 4, 10, 23
- Filter-Methoden*, 28
- Fisher Diskriminante, 30
- Fisher-Diskriminante, 64, 67
- FMEA, 5
- FR, 31
- FTA, 5
- Fuzzy-Klassifikatoren, 56
- FWT, 10
  
- Hörbarkeit, 60
- Hauptkomponentenanalyse, 30
- HMM, 4
- hold-out*, 38
- IAD, 87

- ICE, 9
- ID3, 30
- IG, 33
- iMAD-System, 125
- Informationsgehalt, 33
- Informationsgewinn, 33, 64
- k-Nächste-Nachbarn, 55, 65
- k-NN, 55, 65
- Klassen, 65
- Klassifikationsrate, 8, 19
- KNN, 42
- Kohonen, 95
- Komponentendarstellung, 96
- Korrelationskoeffizient, 31, 64
- Kreuzvalidierung, 39
- Kullback-Leibler-Divergenz, 33
- Kurtosis, Schiefe, 25
- L-Normen, 41
- labeling*, 9
- Lasso, 30
- Lautheit, 26
- Leaky Integration*, 53
- Learning Vector Quantization*, 54
- Least-Mean-Squares*, 44
- Lernende Vektorquantisierung, 54
- Linear-Vector-Quantization*, 65
- LMS, 44
- LVQ, 44, 54, 65
- mADS, 125, 127
- Mahalanobis Distanz, 42
- Markov blanket, 36
- Maximum-Likelihood-Classifier*, 55, 65
- Maximum-Metrik, 42
- Mel-Frequenz-Cepstrum-Koeffizienten, 24
- MFCC, 24
- MI*, 34
- Min-Max-Normalisierung, 63
- Minkowski-Metriken, 41
- ML, 55, 65
- MLP, 65
- MLP-NN, 43, 65
- mobiles akustisches Diagnosesystem, 127
- Modulationsfrequenz, 25
- mRMR, 34
- Multi-Layer Perceptron Neural Network*, 43, 65
- MUSIC*, 126
- Mustererkennung, 37
- Mutual Information*, 34
- n.i.O., 37
- naiver Bayes-Klassifikator, 56
- NB, 56
- neuronale Netze, 42
- neuronale Netztopologien, 43
- Neuronenmodelle, 43
- Normalisierung, 63
- Nulldurchgang-Rate, 25
- Oktavspektrum, 24
- overfitting*, 67
- overfitting*, 28, 65
- PCA, 30
- Perzeptron-Lernregel, 44
- PLM, 3, 4
- psychoakustische Merkmale, 60
- Rauhigkeit, 26
- recheneffiziente Klassifikationsverfahren, 54
- Reduktionsverfahren, 64
- Redundanz, 35
- relative Entropie, 33
- Relief-Algorithmus*, 32
- Relief-Algorithmus*, 64
- repeated hold-out*, 39
- RMS, 24, 62
- rms, 24
- Schärfe, 26
- Scheitelfaktor, 24
- Schnelle Fourier-Transformation, 23
- Schwankungsstärke, 26
- Selektionsverfahren, 27, 64
- self-organized learning*, 37
- Self-Organizing-Maps, 45, 95



SNR, 8  
SOM, 9, 44, 45  
*Spectral Flatness*, 23  
*Spectral Rolloff*, 24  
spektrale Merkmale, 60  
Spektrale Schwankungsstärke, 23  
Spektraler Schwerpunkt, 23  
spezifische Lautheit, 26  
spezifische Schwankungsstärke, 26  
SPL, 26  
*split-sample*, 38  
SQL, 17, 59  
Standardnormalverteilung, 64  
SU, 35  
*subset*, 27, 71, 73  
Super-SAB, 45  
*supervised learning*, 37  
Support-Vector-Machine, 55  
*Support-Vector-Machine*, 65  
SVM, 9, 55, 65  
Symmetrische Unbestimmtheit, 35  
  
Terzspektrum, 24  
Testdatensatz, 38  
Time-Delay Self-Organizing Map, 53  
Tonhaltigkeit, 27  
Trainingsdatensatz, 38  
Transinformation, 34, 64  
*Tree*, 65  
  
*UCI Database*, 65  
  
Wahrscheinlichkeit, 32  
Wavelet, 5  
*Wrapper*-Methoden, 28  
  
z-score-Normalisierung, 63  
z-Transformation, 64  
ZCR, 25

